AN EMBEDDED CALIBRATION TECHNIQUE FOR HIGH-RESOLUTION FLASH TIME-TO-DIGITAL CONVERTERS

by

JAMES CICALO

B.A.Sc., University of British Columbia, 2002

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE

in

THE FACULTY OF GRADUATE STUDIES

(Electrical and Computer Engineering)

THE UNIVERSITY OF BRITISH COLUMBIA

August 2007

Abstract

As CMOS technology continues to advance, device dimensions will continue to decrease, thus enabling the creation of circuits which operate at increasingly greater frequencies. However, this increase in operating frequency has resulted in a reduced tolerance for circuit timing uncertainties. Therefore, techniques capable of measuring the timing characteristics of multi-GHz signals are needed to help address the growing number of timing problems found in modern CMOS circuits. For cost and accuracy reasons, embedded time interval measurement techniques which offer picosecond measurement accuracies and millisecond test-times are required to overcome these challenges.

The "sampling offset" based flash time-to-digital converter (SOTDC) is an embedded time interval measurement technique that has recently garnered much attention due to its attractive properties. These properties include sub-millisecond test times of multi-GHz signals, in addition to the potential for measurement accuracies in the order of picoseconds. However, the accuracy of an SOTDC is strongly dependent upon the capabilities of its calibration technique, and present SOTDC calibration techniques suffer from some very serious limitations. In fact, these limitations are so severe that present calibration techniques are impractical under realistic production test conditions.

This thesis presents the design and analysis of a novel embedded SOTDC calibration technique. The proposed calibration technique offers the potential for both sub-picosecond calibration accuracies and sub-100 millisecond calibration times. However, the main contribution of this work concerns the suitability of the proposed technique with a realistic production test environment. The capabilities of the proposed calibration technique have been proven using both mathematical analysis and behavioural modelling simulations.

Table of Contents

List of Tables

List of Figures

Acknowledgements

I would like to convey my most sincere gratitude to those who have helped me to complete this work. Firstly, I would like to thank my research supervisor, Professor André Ivanov, for providing me with the opportunity to join his research group in addition to the UBC SoC laboratory. Professor Ivanov's continued guidance and support throughout this work was instrumental to its ultimate success. Secondly, I would like to thank my colleagues at the SoC laboratory, and in particular, Zaman Mollah, Andy Kuo, Michael Jones, and Dr. Roberto Rosales. I learned much from our countless discussions, and I will always be grateful for their advice and friendship. I would also like to thank the Canadian Microelectronics Corporation for granting me access to their IC fabrication services, in addition to Micronet, a Canadian Network of Centres of Excellence focused on the design of microelectronic systems, for financial support.

Last, but certainly not least, I would like to thanks my parents, Ken and Teresa, my sister Carolyn, and my fiancée, Linda Zhang, for their immeasurable love and encouragement during the many challenges I encountered over the years.

Chapter 1

Introduction

As CMOS technology continues to advance, device dimensions will continue to decrease, thus enabling the creation of circuits which operate at increasingly greater frequencies. However, this increase in operating frequency has resulted in a decreased tolerance for circuit timing uncertainties. In addition, the behaviour of a circuit, and therefore the timing of its signals, is becoming increasingly sensitive to environmental influences. These environmental influences may disturb the operation of a circuit through a number of mechanisms. These mechanisms include capacitive and inductive coupling, as well as the injection of noise into the power-supply or the substrate of a CMOS circuit [1, 2]. As these mechanisms are becoming increasingly prevalent in modern CMOS circuits, critical path signals are increasingly susceptible to unwanted timing variations.

Unintended timing variations in a signal may cause a circuit to become non-functional. Therefore, the ability to detect, diagnose, and if possible, repair timing problems is of the utmost importance if the reliability of a CMOS circuit is to be guaranteed. However, detecting timing problems in multi-GHz signals can be a very challenging task due to the extremely short time intervals that must be measured. For example, a 10% deviation in the period of a 10 GHz signal translates to a mere 10 ps. Without the ability to detect timing problems in multi-GHz signals, it is not possible to diagnose or repair them. As a result, techniques capable of detecting and diagnosing timing problems in multi-GHz signals are needed to help address the growing number of timing issues found in modern CMOS circuits.

1.1 Time Interval Measurement

The detection or diagnosis of a timing problem in a CMOS circuit is often accomplished with the help of a time interval (TI) measurement technique. TI measurement is a time domain analysis technique that is often used to deduce the timing characteristics of a signal by estimating its threshold crossings in the voltage domain [3]. Many types of TI techniques exist, however they all share a common goal of quantifying the amount of uncertainty in the timing of a signal. Once this timing uncertainty has been quantified, predictions regarding the probability of a circuit's failure can be made. Timing uncertainty is usually referred to as "timing jitter" or "absolute jitter", which is defined as the deviation from the ideal timing of an event, and can be accumulated over many cycles [4, 5]. This definition is illustrated in Figure 1.1, where the amount of timing jitter in a signal under test (SUT) is indicated by the degree of uncertainty in the temporal location of a signal transition.

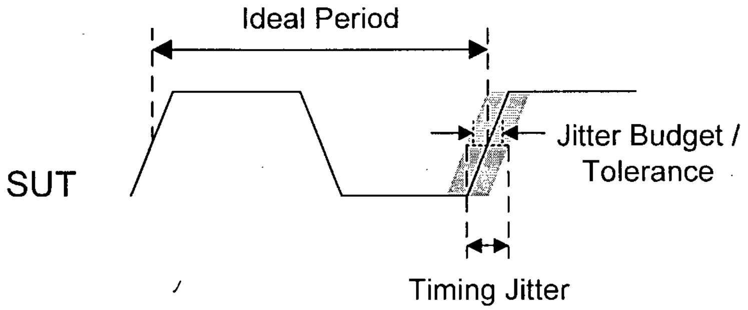


Figure 1.1: Timing jitter in a signal under test (SUT).

Another useful definition that is illustrated in Figure 1.1 is that of the jitter budget or tolerance of a design, which is the maximum amount of timing jitter that can exist in a signal before the circuit fails to operate reliably. Two additional classifications of jitter exist, as illustrated in Figure 1.2. The first of these classifications is the most common of the three, and is known as "period" jitter. Period jitter is simply the deviation of a single period from its ideal value. The second classification is known as "cycle-to-cycle" jitter, and is a measure of the difference between adjacent cycles.

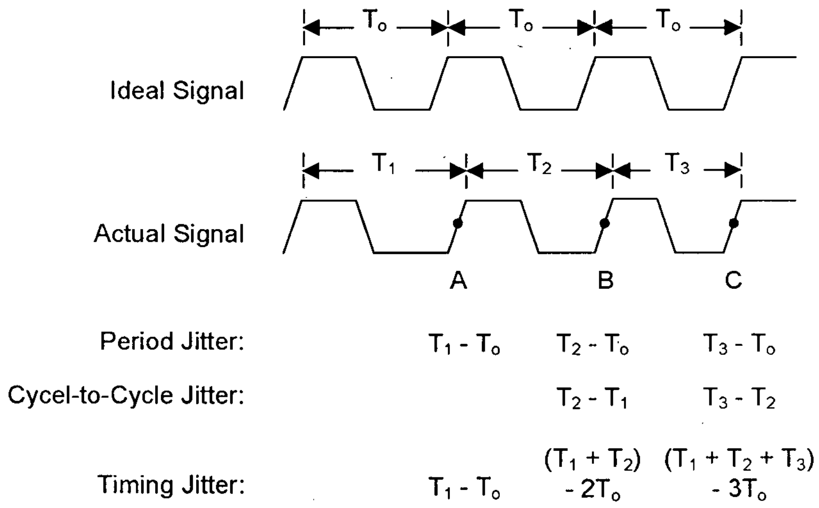


Figure 1.2: Three classifications of jitter.

Each of the aforementioned types of jitter may contain both random and deterministic components, depending upon the source of the jitter. In any case, it is possible to predict the probability with which a signal will exceed a circuit's timing margins by constructing the probability density function (PDF) of the period jitter [6]. The PDF of a purely random source of period jitter is illustrated in Figure 1.3 (a). Inspection of Figure 1.3 (a) reveals that random period jitter can be characterized by a Gaussian distribution. Since a Gaussian distribution is unbounded, its peak-to-peak value (the difference between the shortest and longest cycles) is also unbounded, and is highly dependent upon the number of cycles measured. The PDF of period jitter resulting from both random and deterministic sources is illustrated in Figure 1.3 (b). The shape of this PDF is determined by the convolution of the random and deterministic components' PDFs [7]. As deterministic jitter is bounded in nature, its peak-to-peak value is also bounded.

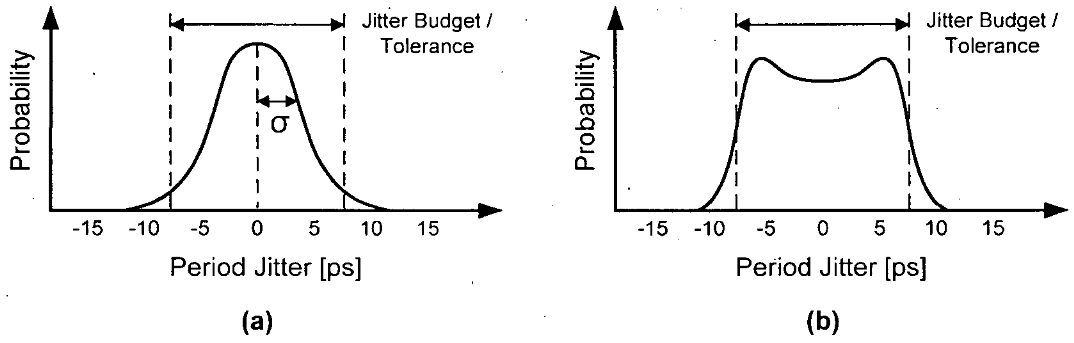


Figure 1.3: PDF of random period jitter (a), and a combination of random and deterministic period jitter (b).

As the function of a time interval measurement technique is to accurately estimate the duration of a time interval, multiple measurements of a signal's period can be performed and subsequently compiled into a histogram. If this histogram is normalized by the number of measurements performed, a PDF of the signal's period jitter can be produced. However, before an accurate PDF can be produced, many cycles need to be measured [8]. This idea is illustrated in Figure 1.4, where the random period jitter of a signal is estimated using three different histograms. Each histogram is drawn using an increasing number of measurement results.

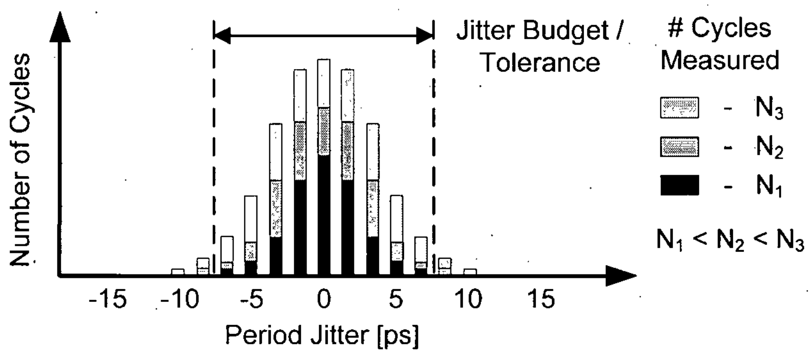


Figure 1.4: The growth of a random period jitter histogram as the number of measurement is increased.

Inspection of Figure 1.4 reveals that both the standard deviation and the peak-to-peak jitter of the histogram may vary as the number of measurement cycles is increased. Although a few thousand measurements are often sufficient to provide an accurate estimate of standard deviation, hundreds of thousands, or even millions of measurements are often required in order to make an accurate prediction of the peak-to-peak jitter in a multi-GHz signal. Such information is frequently used as a metric when determining the probability of circuit failure [8].

While many different time interval measurement techniques exist [9], the choice of which technique to employ for a given application ultimately depends on the measurement requirements. For instance, the measurement of period jitter at giga-bits-per-second (Gbps) data rates necessitates very accurate results, as the jitter budget at these speeds is extremely small. For example, the authors in [20] predict that measurement accuracies of 1 ps or less will be required for bit-error-rate (BER) testing of 10 Gbps integrated circuit (IC) pins.

As previously mentioned, obtaining accurate jitter results may require a large number of measurements. Therefore, as signal data rates increase along with jitter measurement requirements, the total measurement time of TI measurement techniques continues to rise. As a result, only a select group of low test-time measurement techniques are feasible in a volume production test environment, where test time is directly related to product cost [10].

Signal amplitude sampling-based techniques [11] can be used to reconstruct the shape of a voltage-time waveform based on a number of voltage-time samples. While these techniques are not strictly "time interval" based jitter measurement techniques, they have been successfully used to measure jitter with picosecond accuracy [12]. However, signal amplitude sampling-based techniques typically require tens of seconds per measurement, which is far too much time for a volume production test environment [13]. High-frequency production testers can be used to measure jitter with picosecond accuracy in a matter of seconds [14, 15, 16]. However, these testers generally cost millions of dollars. In addition, probing gigaHertz signals for off-chip measurement can introduce significant additional jitter [20]. Therefore, for cost and accuracy reasons, embedded (on-chip) time interval measurement techniques which offer picosecond measurement accuracies and millisecond test-times are very useful tools to enable the cost-effective analysis of a growing number of timing problems found in modern CMOS circuits [17]. In fact, embedded time interval measurement techniques are currently the subject of research within both academia and industry [18].

One new time interval measurement technique which has recently garnered much attention is the "sampling offset" based flash time-to-digital converter (SOTDC) [23]. This time-to-digital converter (TDC) offers sub-millisecond test times for gigaHertz signals, as well as the potential for picosecond measurement accuracies. However, the accuracy of an SOTDC is strongly dependent on the capabilities of its calibration technique. To date, no feasible embedded calibration technique for an SOTDC has been proposed. This thesis is focused on the design of a novel embedded calibration technique for SOTDCs which offers the potential for sub-picosecond calibration accuracies, and calibration times in the order of milliseconds. While specific reference to the calibration of an SOTDC is made, this calibration technique is applicable to any flash-based TDC.

1.2 Thesis Organization

This thesis consists of a total of six chapters. Important background information concerning the evolution of traditional flash-based TDCs into state-of-the-art SOTDCs is presented in Chapter 2. Three previously proposed SOTDC calibration techniques are described in Chapter 3, and the important limitations of each are investigated. Next, the embedded calibration technique proposed in this thesis is described in Chapter 4, followed by an analysis of its capabilities and limitations in Chapter 5. Finally, conclusions regarding the contribution of this thesis are presented in Chapter 6, along with a discussion of future work.

Chapter 2

Flash-Based Embedded Time Interval Measurement Techniques

Embedded time interval measurement can be performed using a variety of techniques, and is often realized using a time-to-digital converter (TDC). A TDC is a circuit that outputs a digital codeword when a time interval is applied to its input, as shown in Figure 2.1. The time interval to be measured, referred to from hereon as T_d, is defined as the difference in time between the rising edge transitions of two signals, which are traditionally referred to as *START* and *STOP*. This digital codeword, once interpreted, approximates the duration of the time interval.

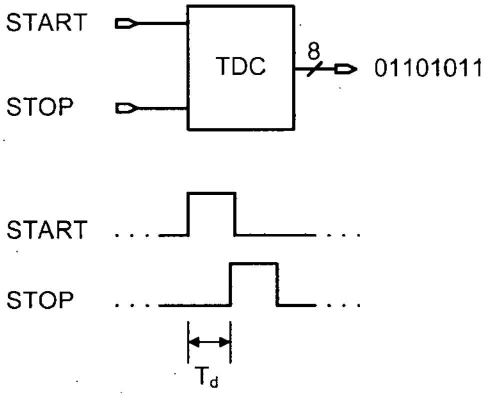


Figure 2.1: The role of a TDC.

While many different types of TDCs exist, they can all be evaluated against the following criteria:

- **Accuracy**:

  How closely the interpreted digital codeword matches T_d.

- **Resolution**:

  The smallest measurable difference in T_d.

- **Precision**:

  The degree to which a set of measurements of the same T_d agree.

- **Measurement rate**:

  The maximum rate at which different T_ds can be applied to the TDC's input while still receiving correct codewords at its output.

- **Dynamic range**:

  The ratio of the maximum to minimum T_d measurable by the TDC.

- **Power and Area requirements**:

  The area required to implement an on-chip TDC with certain accuracy, resolution, precision, measurement rate, and dynamic range specifications, in addition to the power consumed by this TDC.

As the accuracy, resolution, precision, dynamic range, and measurement rate requirements placed upon TDCs become increasingly stringent, trade-offs are necessary in order to construct a feasible TDC architecture. Many TDC architectures target a reduced measurement rate in order to meet the accuracy, resolution, and precision requirements. Examples of such TDCs include the Vernier oscillator-based TDC [19], and the undersampling-based TDC described in [20]. However, this trade-off can be very costly for integrated circuit (IC) manufacturers, since the resulting increase in production test time increases overall production costs.

This chapter examines the evolution of "flash" TDC architectures. The chapter begins with a description of the most primitive form of a flash TDC, and concludes with a presentation of the state-of-the-art in flash TDC design, where picosecond measurement resolutions are achievable. In general, Flash TDCs are capable of very high measurement rates. In fact, flash TDCs are capable of operating at or near the frequency of the signal or signals under test, from which the START and STOP signals are derived. Flash TDCs are analogous to flash analog-to-digital converters (ADCs), since their output codeword is determined in a single step by a bank of comparators [21]. Therefore, the flash TDC architecture is a very good candidate for embedded time interval measurement in both a production test environment or in a customer application, where measurement time is of comparable importance to measurement accuracy, resolution, and precision.

2.1 Single Delay Line-Based Flash TDC

The most basic form of a flash TDC is the single delay line-based flash TDC, which is illustrated in Figure 2.2. This TDC architecture has two primary inputs, namely *START* and *STOP*, and a multitude of outputs, labelled C_1 to C_N in this embodiment.

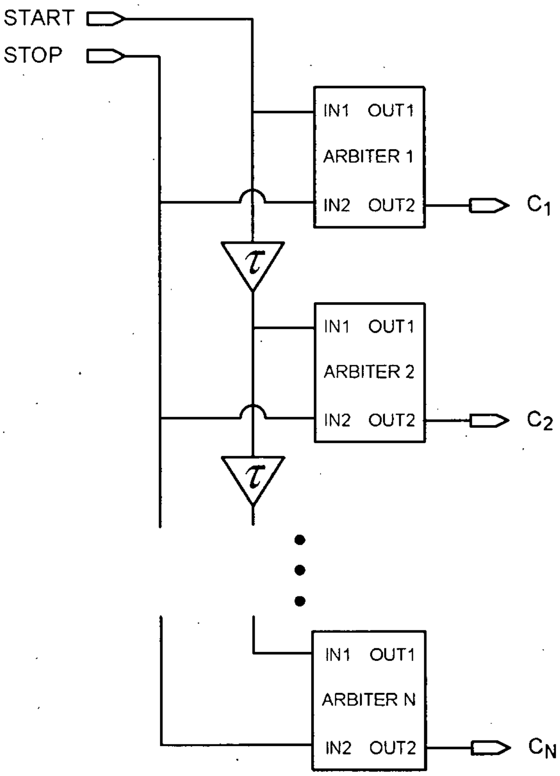


Figure 2.2: Single delay line-based flash TDC.

If we define the instant at which the *START* signal transitions from a low to a high logic level as t_{start}, and if we define t_{stop} analogously for the *STOP* signal, then we can describe T_d in mathematical terms with the following equation:

$$T_d = t_{stop} - t_{start} \tag{2.1}$$

As shown in Figure 2.2, the *START* signal is delayed by a single buffer as it propagates from one arbiter to the next. The delay of each buffer is equal to τ. At each stage, an arbiter determines which of its two inputs was the first to transition from a low to a high logic level, i.e., the first to make a "positive" transition. If *IN1* is the first to perform such a transition then *OUT1* is set to a high logic level and *OUT2* to a low logic level, and vice versa if *IN2* is the first to arrive. Figure 2.3 illustrates the operation of a single delay line-based flash TDC consisting of 4 arbiters. This type of TDC can be referred to as a 4-bit single delay line-based flash TDC.

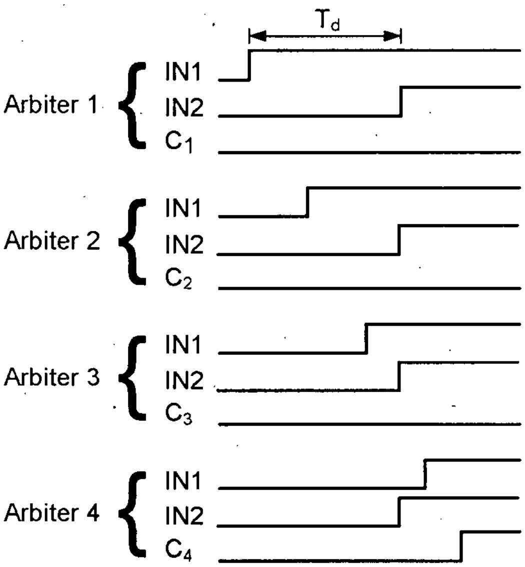


Figure 2.3: Single delay line-based flash TDC timing waveform.

As is shown in Figure 2.3, a single delay line-based flash TDC produces a thermometer code digital output ($C_4C_3C_2C_1$ = 1000). T_d can be approximated by noting the location of the "0" to "1" transition in the output codeword. In the above example, T_d is shown to satisfy the following condition:

$$2\tau < T_d \leq 3\tau \qquad (2.2)$$

The resolution of this TDC is limited by the buffer delay, τ. This buffer delay has a practical lower bound due to the physical constraints of the technology in which it is implemented. Therefore, for high-resolution applications, a single delay line-based flash TDC may be inadequate.

2.2 Vernier Delay Line-Based Flash TDC

In order to overcome the resolution limitations of a single delay line-based flash TDC, a second delay line can be added, as shown in the Vernier delay line-based flash TDC of Figure 2.4.

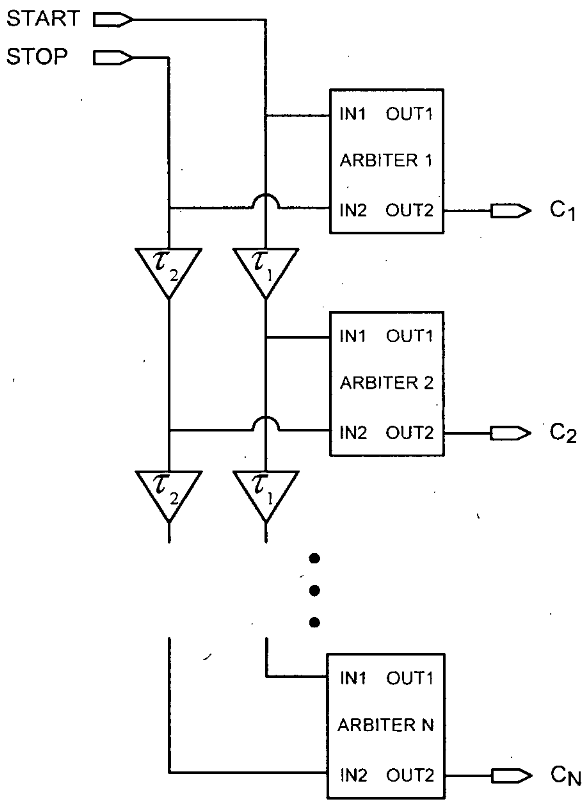


Figure 2.4: A Vernier delay line-based flash TDC.

This second delay line is used to incrementally delay the *STOP* signal as it propagates from one arbiter to the next, as is done to the *START* signal in the single delay line-based flash TDC. The delay of each buffer in the *START* signal path is equal to τ_1, whereas the delay of each buffer in the *STOP* signal path is equal to τ_2. An example of the method of operation of a 4-bit Vernier delay line-based flash TDC is illustrated in Figure 2.5.

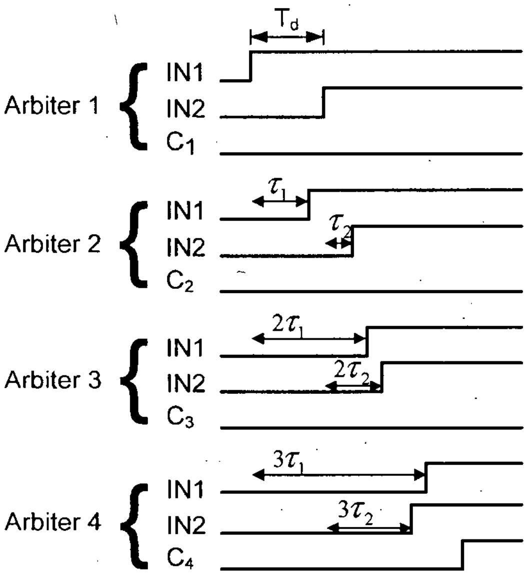


Figure 2.5: Vernier delay line-based flash TDC timing waveform.

As shown in Figure 2.5, a Vernier delay line-based flash TDC also produces a thermometer code digital output ($C_4C_3C_2C_1$ = 1000). T_d can be found using the same procedure described for the single delay line-based flash TDC, i.e., by noting the location of the "0" to "1" transition in the output codeword. In the example illustrated in Figure 2.5, T_d is shown to satisfy the following condition:

$$2(\tau_1 - \tau_2) < T_d \leq 3(\tau_1 - \tau_2) \tag{2.3}$$

The buffer delay difference, i.e., τ_1 - τ_2, where $\tau_1 > \tau_2$, defines the resolution of a Vernier delay line-based flash TDC. Therefore, sub-gate delay resolution can be achieved with this architecture.

Calibration of a Vernier delay line-based flash TDC is done to ensure that the buffers in each of the two delay lines provide the required delay, i.e., τ_1 or τ_2. Normally a delay-locked-loop (DLL) is used to accomplish this, ensuring that integral nonlinearity (INL) errors in the converter are minimized [22]. However, the arbiters are most often constructed from flip-flops, since a flip-flop is essentially an arbiter. While flip-flops make efficient arbiters, their non-zero setup times may influence the buffer delay difference (τ_1 - τ_2) and hence contribute to the TDC's differential nonlinearity (DNL) error, as a DLL-based calibration technique cannot be used to perform stage-by-stage calibration. For example, there is no impact on the measurement accuracy of the TDC as long as the flip-flops have identical setup times, in which case they can be treated as a constant and removed from the measurement results. However, the setup times of flip-flops on the same semiconductor die can vary significantly due to process variations. For example, variations as large as 50 ps have been observed in a 0.35 μm CMOS process [23]. As the resolution of a Vernier delay line-based flash TDC is increased, the importance of a flip-flop's setup time is amplified, as it is not accounted for during calibration. Therefore, there exists a limit to how small the buffer delay difference can be made before the variability between flip-flop setup times begins to add a significant level of error to the measurement results. For the measurement of 5 and 10 Gbps data rate signals, where the required accuracy is 10 ps or better, a Vernier delay line-based flash TDC is inadequate [20].

2.3 Sampling Offset-Based Flash TDC

A novel concept discussed in [23] attempts to address the time interval measurement accuracy requirements of 10 Gbps data rate signals and beyond. The author in [23] suggests that a TDC with a resolution of 2 ps or less can be constructed by removing the buffers from a Vernier delay line-based flash TDC, thereby making use of the inherent variations in the setup times of the arbiters. This type of TDC, shown in Figure 2.6, is known as a "sampling offset" TDC (SOTDC).

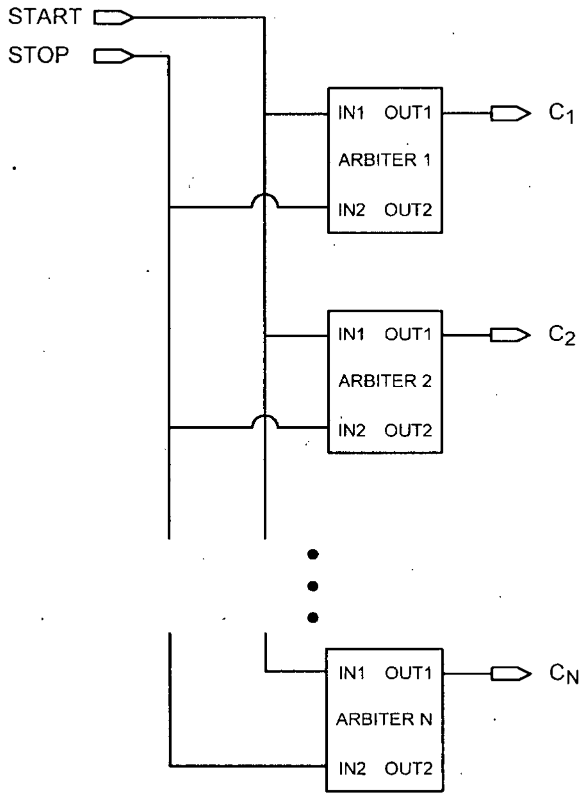


Figure 2.6: Sampling offset-based flash TDC.

The term "sampling offset TDC" arises from the fact that a time interval is quantized using the difference in the setup times, or sampling offsets, of the arbiters, assuming they are known. This is in contrast to a Vernier delay line-based TDC, which uses a difference in buffer delays to quantize time. Instead of implementing the arbiters or "sampling elements" with flip-flops, the author in [23] chose to use symmetric CMOS arbiters. A symmetric CMOS arbiter schematic is drawn in Figure 2.7. This circuit arbitrates between two inputs, *IN1* and *IN2*, by determining which input was the first to perform a low to high transition, i.e., a positive transition.

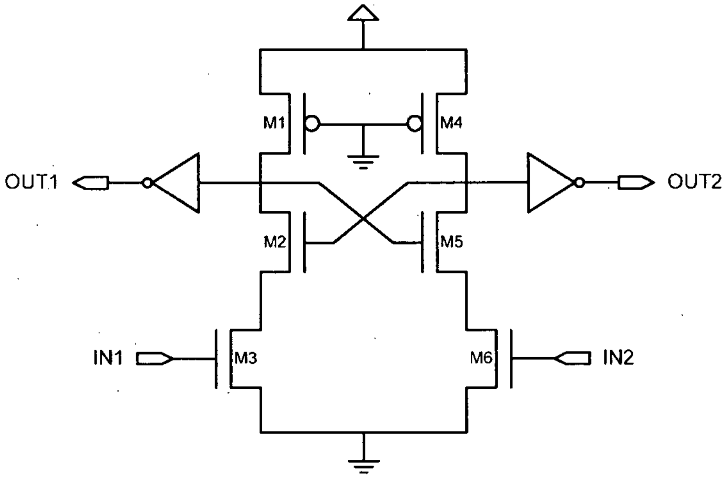


Figure 2.7: Symmetric CMOS arbiter.

The operation of a symmetric CMOS arbiter relies on the use of positive feedback. With inputs *IN1* and *IN2* discharged low, transistors M3 and M6 remain in cutoff mode. Therefore, the drains of M2 and M5 remain precharged high through transistors M1 and M4, and outputs *OUT1* and *OUT2* remain predischarged low. Now, if *IN1* is the first input to perform a positive transition, current will flow down the left-hand side of the arbiter as M3 leaves the cutoff mode of operation. If M1, M2, and M3 are properly sized, the voltage on the drain of M2 will fall low enough to cause *OUT1* to switch high. In addition, since the drain of M2 is connected to the gate of M5, M5 will enter the cutoff region, in turn maintaining a high voltage at the drain of M5 and a low voltage at *OUT2*. It is this use of positive feedback between transistors M2 and M5 that allows this arbiter to successfully resolve picosecond-timing differences, as demonstrated in [24].

An arbiter such as the one illustrated in Figure 2.7 is said to be perfectly symmetric if its left hand side behaves identically to its right hand side. As a consequence of this perfect symmetry, its sampling offset (t_{so}) is equal to zero seconds. An arbiter with a non-zero sampling offset is said to be "biased" towards one of its inputs. Therefore, a perfectly symmetric arbiter does not exhibit a bias towards either input. As a result, the first input to transition from a low to a high logic level is always recorded as such, with the corresponding output set to a high logic level. This behaviour is illustrated in Figure 2.8.

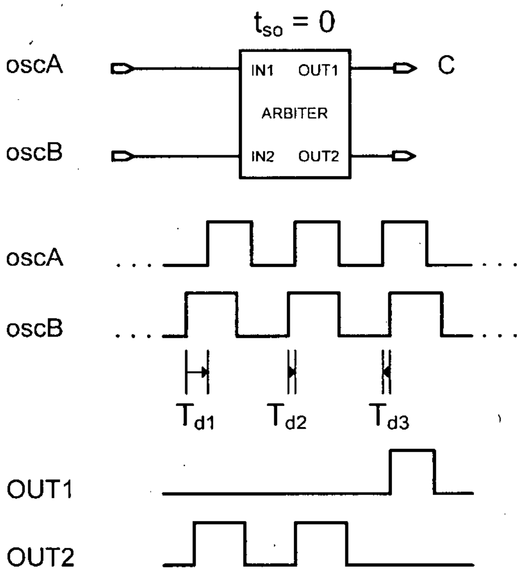


Figure 2.8: Behaviour of a perfectly symmetric arbiter.

In Figure 2.8, two oscillators, *oscA* and *oscB*, are depicted as the inputs to a perfectly symmetric arbiter. The frequency of *oscA* is slightly greater than that of *oscB*. Therefore, a sequence of varying time intervals is generated from the rising edge transitions of the two oscillators. If the temporal location of a rising edge transition of *oscB* is denoted as $t_{oscB(i)}$, and if $t_{oscA(i)}$ is defined analogously for *oscA*, then each time interval can be expressed mathematically as:

$$T_{d(i)} = t_{oscB(i)} - t_{oscA(i)} \tag{2.4}$$

The preceding definition allows for the sampling offset of the arbiter in Figure 2.8 to be bound by the following inspection-based equation:

$$T_{d2} < t_{so} \leq T_{d3} \tag{2.5}$$

A biased arbiter, however, exhibits a non-zero sampling offset. This bias can be the result of transistor mismatches between the left and right hand sides of the arbiter, and is often attributed to process variations. However, it can be useful to intentionally bias an arbiter, in which case the transistor mismatches are the result of design intent [25]. The behaviour of a biased arbiter is illustrated in Figure 2.9, where a buffer delay (τ_{del}) has been inserted before input *IN1* of a perfectly symmetric arbiter in order to mimic the behaviour of a biased arbiter.

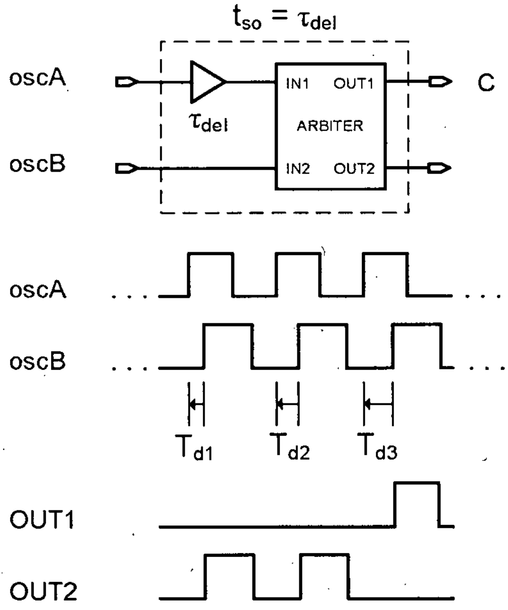


Figure 2.9: Behaviour of a positively biased arbiter.

Analogously, the sampling offset of the arbiter in Figure 2.9 can be bound with the following inspection-based equation:

$$T_{d2} < t_{so} \leq T_{d3} \tag{2.6}$$

In summary, inserting a buffer before input *IN1* of a perfectly symmetric arbiter results in an arbiter that is biased by an amount equal to the delay of the buffer (τ_{del}). For this reason, this type of arbiter is known as a “positively biased” arbiter.

A typical SOTDC can be constructed from several positively biased arbiters, each with a unique sampling offset. If the arbiters are positioned within the SOTDC in order of smallest t_{so} to largest, then the output codeword will be in the form of a thermometer code.

Therefore, if a time interval, T_d, where $T_d > 0$, is applied to the SOTDC, then the value of T_d can be approximated by noting the location of the "0" to "1" transition in the output codeword. For example, assume an SOTDC consisting of only 4 positively biased arbiters exists. Now, if a T_d which is greater than the sampling offset of arbiters 1 and 2, but smaller than that of arbiters 3 and 4, is applied to the SOTDC, then the output codeword will look as follows: $C_4C_3C_2C_1 = 0011$. Such an output codeword can be used to approximate the value of the applied time interval, T_d, as shown in Equation (2.7), where t_{so2} and t_{so3} represent the sampling offsets of arbiters 2 and 3, respectively.

$$t_{so2} \leq T_d < t_{so3} \tag{2.7}$$

If the sampling offsets of the arbiters are equally spaced, then the error in the above approximation must be bounded by the resolution of the SOTDC, which is defined as the step size of the arbiter sampling offsets. The challenge associated with using an SOTDC for time measurement lies in determining the sampling offsets of the arbiters, as without such information it is impossible to extract useful data from the arbiter outputs. Several calibration techniques have been developed in order to measure the sampling offsets of the arbiters within an SOTDC. The merits and drawbacks of each are presented in the following chapter, and a new calibration technique is proposed in Chapter 4.

Chapter 3

Embedded Calibration of a Sampling Offset-Based Flash TDC

Present SOTDC calibration techniques suffer from some very serious limitations. The most straightforward of these techniques require an accurately known sequence of closely spaced T_d values, which for a picosecond resolution SOTDC is very difficult to generate on-chip. A more sophisticated technique, as described in [23], requires precise knowledge of the mean sampling offset of the SOTDC arbiters. Unfortunately, such information is usually not available. Another technique, as described in [26], requires the use of an external signal generator or an on-chip DLL in order to generate T_d values which are not necessarily closely spaced, but accurately known nonetheless. Time interval accuracies in the order of picoseconds are required for successful implementation of this technique, and therefore dictate the use of only the most accurate signal generators or on-chip DLLs. In addition, this technique employs two on-chip variable delay elements that must be calibrated with picosecond accuracy in order to resolve any skew introduced between the output of the T_d

generator and the input of the SOTDC, and is therefore not a complete solution. The technique proposed in Chapter 4 is exempt from any of these deficiencies.

3.1 Behaviour of a Non-Ideal Arbiter

Before the aforementioned calibration techniques can be fully understood, a model that incorporates thermal noise in an arbiter must be developed. Such a model has been reported in [26]. This model suggests that the sampling offset of an arbiter is not a fixed number, but should instead be treated as a random variable that changes with time. The preceding implies that the sampling offset of an arbiter at a particular instant in time can only be described as having a certain probability of being a particular value. This "instantaneous" sampling offset is denoted as t_{iso}.

3.1.1 A Model of Thermal Noise in an Arbiter

An ideal arbiter is assumed to have a deterministic output, i.e., the arbiter's output can be predicted exactly if its input is known. Therefore, a given T_d will produce a consistent output from an ideal arbiter. However, as discussed in [23, 27, 28], arbiters implemented using CMOS circuit elements are not ideal, and therefore do not behave deterministically. For example, thermal noise generated in the circuit elements of an arbiter can induce nondeterministic behaviour. A model which illustrates the impact of thermal noise in an arbiter has been developed in [29] and is illustrated in Figure 3.1.

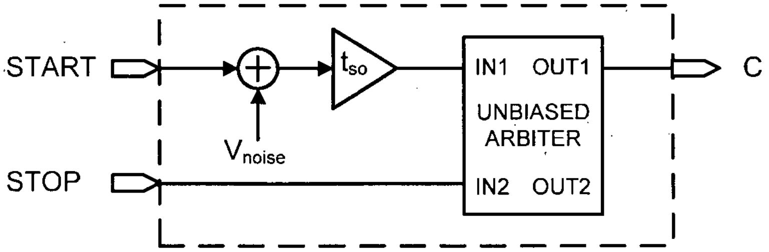


Figure 3.1: Voltage domain model of thermal noise in a biased arbiter.

V_{noise} is a source of noise in the voltage domain, and is the result of thermal noise within the arbiter's circuit elements. This noise is assumed to be white Gaussian noise, with a standard deviation of σ_v and a mean of zero. However, a time domain model of thermal noise in an arbiter is more useful for time interval measurement purposes, since with such a model it is possible to account for the impact of thermal noise in the time domain. A time domain model has been developed in [23, 26], and is illustrated in Figure 3.2.

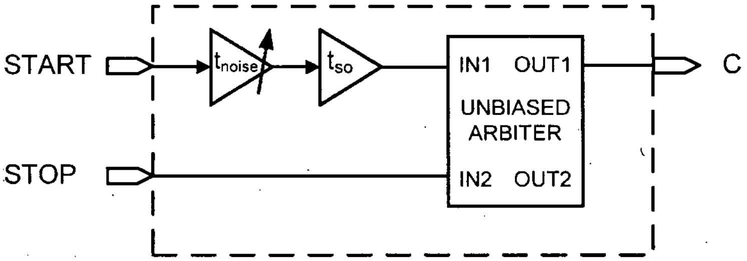


Figure 3.2: Time domain model of thermal noise in a biased arbiter.

With the time domain model, V_{noise} has been replaced with t_{noise}, which functions as a variable delay element. A linear relationship between V_{noise} and t_{noise} is assumed in [23, 26], which allows for t_{noise} to be described by a Gaussian probability density function (PDF), with a standard deviation of σ_t and a mean of zero. Therefore, the time domain model of thermal noise in an arbiter states that the sampling offset of an arbiter is not a single number, but rather a distribution of numbers that can be described with a Gaussian PDF. The mean of this distribution is t_{so} and the standard deviation is σ_t. The sampling offset of an arbiter according to the time domain model of thermal noise in a biased arbiter is depicted in Figure 3.3.

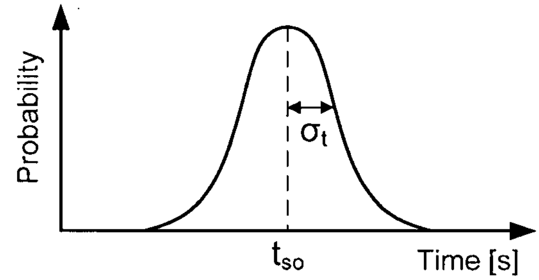


Figure 3.3: PDF of the sampling offset of a biased arbiter taking into account thermal noise.

If the Gaussian PDF shown in Figure 3.3 is integrated over time, the Gaussian cumulative density function (CDF) is produced. This is a useful function since it specifies the probability with which t_{iso} is less than or equal to a specific temporal value, as shown in Figure 3.4, where the temporal value of interest is T_d.

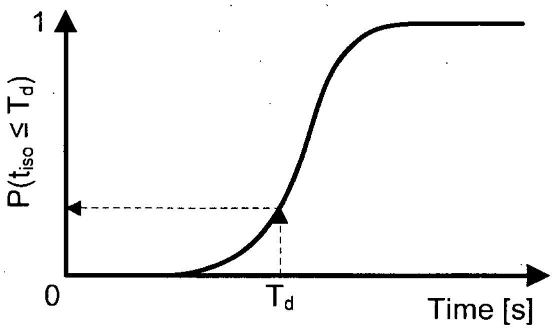


Figure 3.4: Gaussian CDF.

Therefore, according to the time domain model of thermal noise in an arbiter, the probability that a given T_d is greater than or equal to the sampling offset of an arbiter is given by the Gaussian CDF:

$$P(C=1) = P(t_{so} \leq T_d) = \frac{1}{2}\left[1 + erf\left(\frac{T_d - t_{so}}{\sigma_t \sqrt{2}}\right)\right] \tag{3.1}$$

where erf(x) is the "error function", encountered when integrating a normalized Gaussian function [23, 30].

It is interesting to note that the mean sampling offset of an arbiter, t_{so}, can be found from either the PDF or the CDF of the arbiter's sampling offset. Using the PDF of the arbiter's sampling offset, t_{so} can be calculated by finding the mean of the distribution. The CDF of the arbiter's sampling offset can be used to find t_{so} by estimating the value of t that satisfies $P(t_{iso} \leq t) = 0.5$, as illustrated in Figure 3.5.

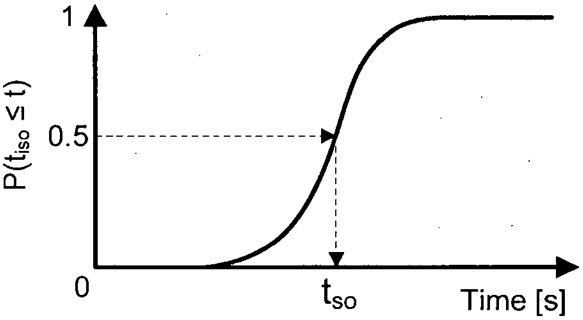


Figure 3.5: Calculation of t_{so} from the CDF of the sampling offset of a non-ideal arbiter.

Of course, if both t_{so} and σ_t are known, then the PDF and the CDF of the arbiter's sampling offset are easily reproduced.

3.1.2 Non-Ideal Arbiters and Time Interval Measurement

An interesting observation concerning the sensitivity of an arbiter to time intervals near t_{so} can be explained with the use of Figure 3.6. The aforementioned figure depicts the response of a symmetric CMOS arbiter with thermal noise to various time intervals. However, the x-axis in this figure has been altered to emphasize the extent to which the arbiter's output can vary with respect to the standard deviation of the thermal noise.

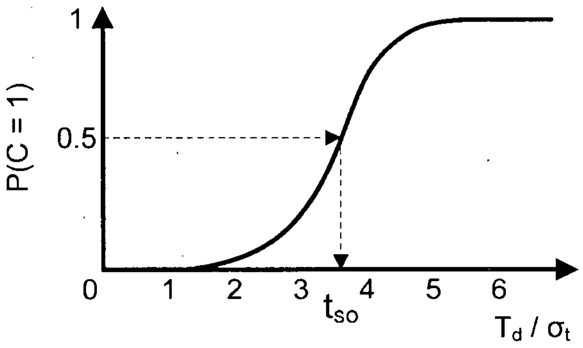


Figure 3.6: Sensitivity of the output of an arbiter to σ_t.

From Figure 3.6 it can be seen that the output of an arbiter exhibits a strong sensitivity to time intervals near the arbiter's mean sampling offset, t_{so}. In fact, a time interval equal to t_{so} - $3\sigma_t$ almost always elicits a different response from the arbiter than one equal to t_{so} + $3\sigma_t$. Therefore, the sensitivity of such an arbiter is highly dependent upon the standard deviation of the thermal noise, σ_t. A test chip consisting of a 64-bit SOTDC has been fabricated in a 0.35 μm CMOS, and is described in [23]. Measurements from this test chip report a σ_t of 0.35 picoseconds. This number suggests that a symmetric CMOS arbiter is suitable for time interval measurement when picosecond accuracy is required.

3.2 Direct Calibration Technique

One very intuitive method to calibrate a sampling offset based flash TDC is to input a sequence of increasing time intervals (T_ds) into the SOTDC, beginning with a known time interval. Each T_d should differ from its predecessor by a constant amount of time, denoted as t_Δ. The instantaneous sampling offset of each arbiter can then be estimated using the value of the first T_d to produce a positive transition at the arbiter's output. The calibration of an 8-bit SOTDC is illustrated in Figure 3.7. In this figure, two oscillators of slightly different frequency, denoted as *oscA* and *oscB*, generate the sequence of T_ds.

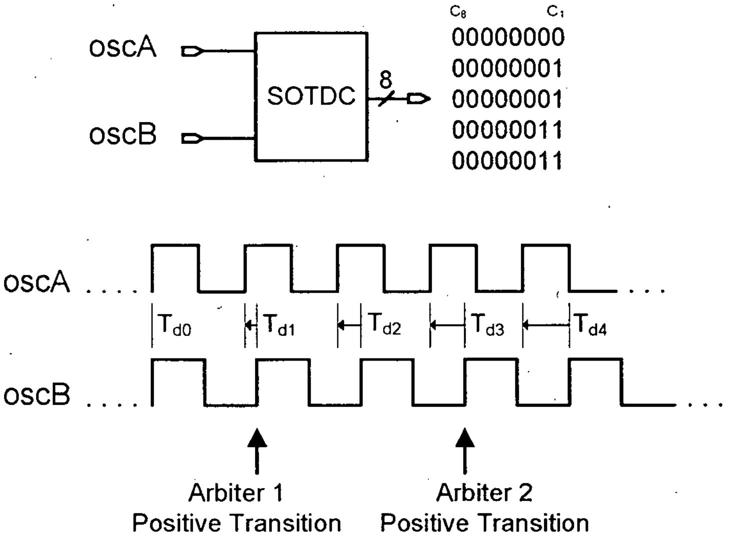


Figure 3.7: Direct SOTDC calibration technique.

The SOTDC shown in Figure 3.7 produces an 8-bit codeword for each T_d, and this codeword is generated from the concatenation of the arbiter outputs, C_8 – C_1. Normally the arbiters are positioned in order of smallest t_{so} to largest. Under such a scenario, if σ_t is much less than the difference in the sampling offsets of adjacent arbiters, then a thermometer code can be expected at the SOTDC output. As previously mentioned, the instantaneous sampling offset of an arbiter within the SOTDC can be estimated using the first T_d from the sequence of time intervals to produce a positive transition at the arbiter's output, as shown in Figure 3.8.

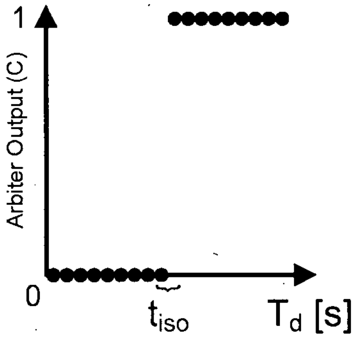


Figure 3.8: Response of an arbiter to a sequence of increasing time intervals.

Mathematically, the t_{iso} of an arbiter can be bound with the following equation:

$$T_{d(i-1)} < t_{iso} \leq T_{d(i)} \tag{3.2}$$

where $T_{d(i)}$ indicates the first T_d to produce a positive transition at the arbiter's output, and $T_{d(i-1)}$ indicates its predecessor. This equation can be rewritten as follows:

$$T_{d(i)} - t_{\Delta} < t_{iso} \leq T_{d(i)} \tag{3.3}$$

Therefore, a reasonable estimate of t_{iso} is:

$$t_{iso} = T_{d(i)} - t_{\Delta}/2 \tag{3.4}$$

The error in this estimate of t_{iso} is bound by ± $t_{\Delta}/2$.

While it is useful to know t_{iso}, the real objective of any SOTDC calibration technique is to determine the mean sampling offset of an arbiter, t_{so}. Therefore, repeating the process depicted in Figure 3.7 multiple times may yield different yet useful results, as illustrated in Figure 3.9.

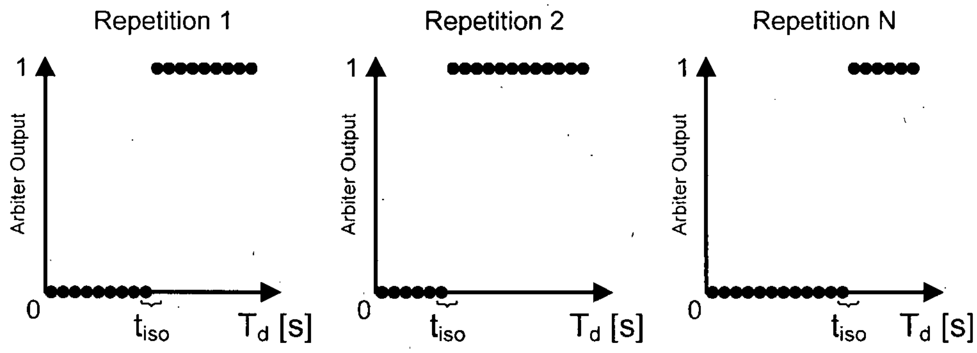


Figure 3.9: Response of an arbiter to several repetitions of a sequence of increasing T_ds.

A histogram of an arbiter's response to N repetitions of a sequence of T_ds can be plotted by summing the number of times the arbiter's output (C) is a logic '1' for each T_d, as shown in Figure 3.10. Since the variation in t_{iso} follows a Gaussian PDF, the histogram has the shape of a Gaussian CDF, assuming a sufficient number of repetitions have been performed. Now, if the histogram data is normalized and an appropriate curve fitting function is used, such as a cubic spline function, a Gaussian CDF may be produced. From this CDF the mean sampling offset of the arbiter (t_{so}) can be determined. This is accomplished by finding the point on the CDF curve for which the arbiter's output is a logic '1' exactly half the time. The temporal value that corresponds to this point is the estimated mean sampling offset of the arbiter, or t_{eso}.

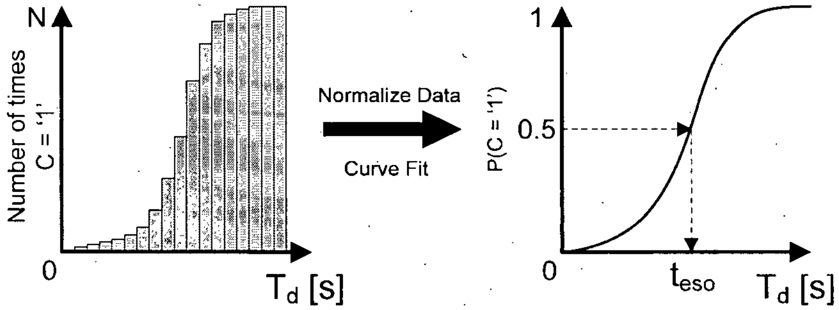


Figure 3.10: Histogram and CDF of the output of an arbiter.

In order to produce an accurate Gaussian CDF from an arbiter's response to a sequence of increasing time intervals, t_{Δ} needs to be chosen carefully, as will be discussed in the following section.

3.2.1 Analysis

As noted earlier, experimental results from a 64-bit SOTDC fabricated in a 0.35 µm CMOS process indicate that the standard deviation of the thermal noise in an arbiter is approximately 0.35 picoseconds [23]. This result places an important bound on the size of t_{Δ}. If the chosen t_{Δ} is approximately equal to or less than σ_t, then the histogram and the resulting CDF constructed from the data collected during arbiter offset calibration will closely resemble those shown in Figure 3.10. To further explain, Figure 3.11 may be of use. In this figure, the time intervals used during calibration are plotted on the x-axis of the arbiter sampling offset PDF. From this PDF a histogram of the arbiter's output for each T_d input is drawn. This histogram is drawn with the assumption that the number of repetitions (N) is large enough to ensure that the collection of arbiter instantaneous sampling offsets produces an arbiter sampling offset PDF that is nearly Gaussian.

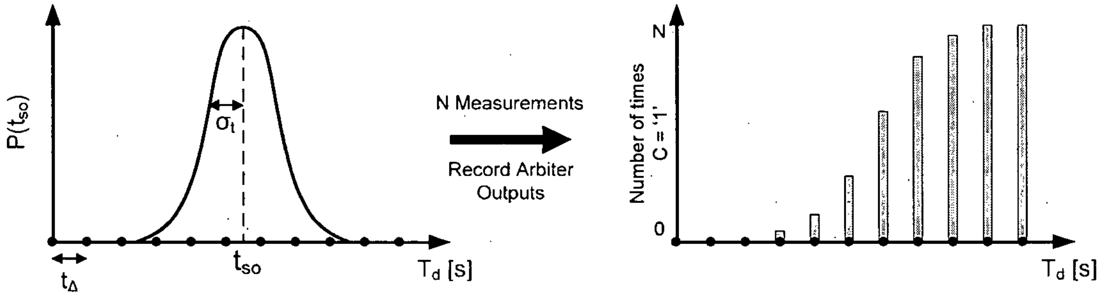


Figure 3.11: Histogram of the output of an arbiter when $t_\Delta \leq \sigma_t$.

The key observation to be made here is that since t_Δ is approximately of the same magnitude as σ_t, a histogram which closely resembles a Gaussian CDF can be drawn, and curve fitting of this histogram to find the arbiter's sampling offset, as shown in Figure 3:10, can be done with reasonable accuracy.

However, if the chosen t_Δ is too large, then the histogram may resemble the one shown in Figure 3.12.

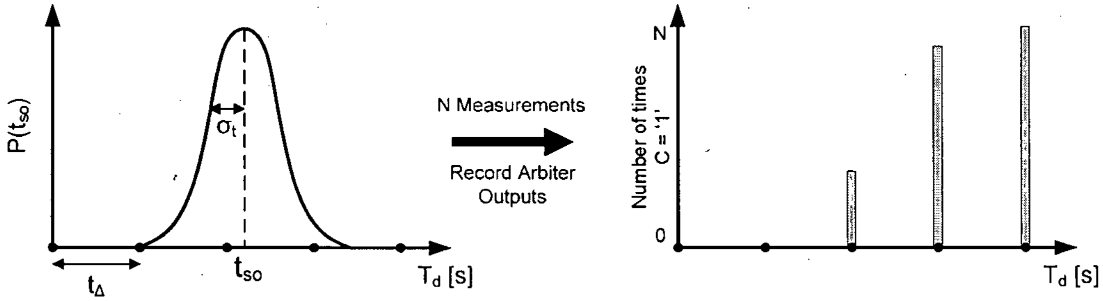


Figure 3.12: Histogram of the output of an arbiter when $t_\Delta > \sigma_t$.

It can be observed from Figure 3.12 that there exists only a small number of useful data points to which a curve can be fitted. Applying a curve fitting function to a small number of data points inevitably leads to an error in the estimation of t_{so} that is much larger than would otherwise be obtainable if t_{Δ} had been properly chosen. The error in the estimation of t_{so}, known as the calibration error, is defined as:

$$t_{ce} = t_{eso} - t_{so} \tag{3.5}$$

As part of this thesis, a quantitative analysis of the relationship between t_{Δ} and t_{ce} has been performed using a software model of the direct calibration technique, one that accounts for thermal noise in an arbiter. This model has been constructed using Matlab, and accepts t_{Δ} and σ_t as parameters, in addition to the t_{so} of each arbiter in the array. Using this information, the model constructs a sequence of T_ds that are applied to the inputs of the array of arbiters. It is also possible to specify the number of repetitions (N) of the sequence of T_ds via an additional parameter. The output of this model is the root-mean-square (RMS) value of t_{ce} for the calibrated array of arbiters, estimated using a 6-th order polynomial fit of the arbiter output histograms.

In order to ascertain the capabilities of the direct calibration technique over a range of t_{Δ} values, several simulations were performed using the aforementioned model. In each case the array was specified to be 100 arbiters long, and N was varied incrementally in powers of 10, beginning at 100 and ending at 1 000 000. In order to keep the results independent of σ_t, t_{Δ} and the RMS value of t_{ce} are expressed in terms of σ_t. The results of eleven simulations for five different values of N are shown in Figure 3.13.

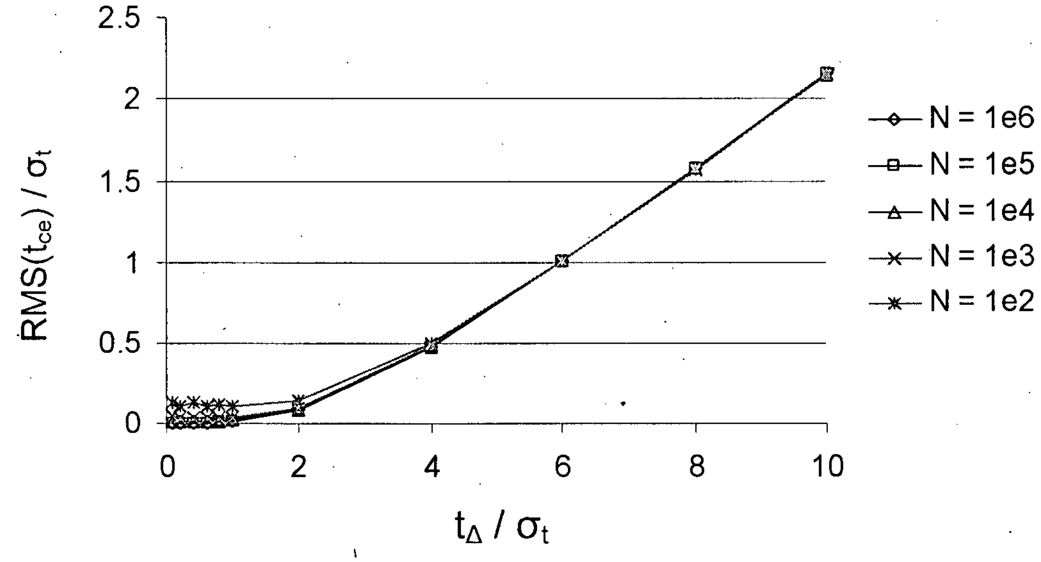


Figure 3.13: RMS t_{ce}/σ_t vs. t_Δ/σ_t using the direct calibration technique.

From Figure 3.13 it can be observed that if $t_\Delta \leq 6\sigma_t$, then the RMS value of t_{ce} is approximately bounded by σ_t, for any value of N. However, as t_Δ continues to increase, so does the RMS value of t_{ce}. In fact, for $t_\Delta \geq 4\sigma_t$, the RMS value of t_{ce} increases linearly with t_Δ. By the time t_Δ reaches $10\sigma_t$, the RMS value of t_{ce} has already surpassed $2\sigma_t$. For an SOTDC with a resolution of 1 ps, i.e., the sampling offset of each arbiter differs from that of its neighbours by 1 ps, an RMS t_{ce} equal to σ_t may be tolerable, assuming $\sigma_t \approx 0.35$ ps. However, an RMS t_{ce} equal to $2\sigma_t$ (0.7 ps) may not be tolerable. Therefore, if such an SOTDC is calibrated using the direct calibration technique, the required t_Δ may be less than 3.5 ps. This requirement may not be practical, as the accurate generation of known time intervals with picosecond temporal resolution is very difficult to achieve on-chip.

One intriguing question which to this point has remained unanswered is the quantitative effect of thermal noise on the accuracy of the arbiter sampling offset estimations obtained using the direct calibration technique. This question can be answered by comparing the RMS t_{ce} from Figure 3.13 with the theoretical RMS t_{ce} of a noise-free arbiter that is calibrated using the direct calibration technique. To calculate the theoretical RMS t_{ce} of the direct calibration technique, the standard RMS formula for a continuous distribution, as shown in Equation (3.6) [31], may be used.

$$RMS(t_{ce}) = \sqrt{\frac{\int P(t_{ce}) t_{ce}^{2} dt_{ce}}{\int P(t_{ce}) dt_{ce}}} \tag{3.6}$$

To solve Equation (3.6), the limits of integration must be determined. In order to determine the limits of integration, the curve fitting procedure used in the direct calibration technique to estimate the t_{so} of a noise-free arbiter must be understood.

The response of a noise-free arbiter to a sequence of T_ds is shown in Figure 3.14. As shown in this figure, the histogram of this response resembles a discrete-time step function. Since there are only two useful data points to which a curve can be fitted, the most sensible approach is to linearly interpolate between the two points in order to construct the CDF of the arbiter's sampling offset and approximate t_{so}.

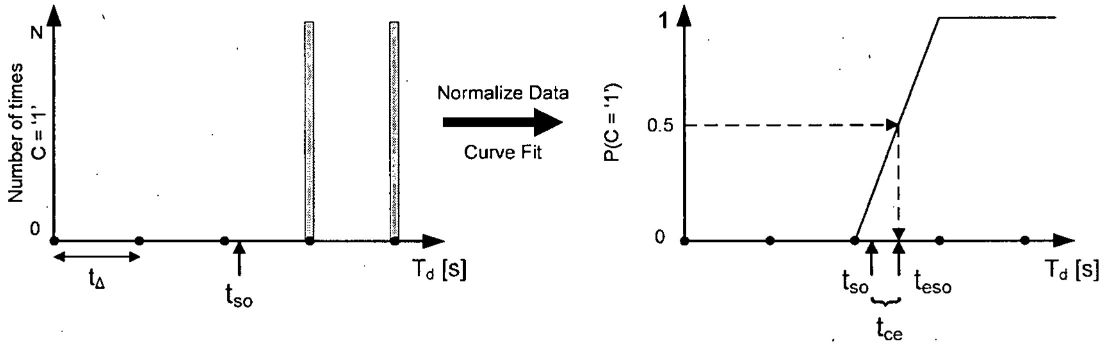


Figure 3.14: Histogram and CDF of the output of a noise-free arbiter.

The error in this approximation, t_{ce}, can be described mathematically by recognizing that the sampling offset of a noise free arbiter is estimated as:

$$t_{eso} = (T_{d(i)} + T_{d(i-1)})/2 = T_{d(i)} - t_{\Delta}/2 \tag{3.7}$$

where $T_{d(i)}$ indicates the first T_d to produce a positive transition at the arbiter's output, $T_{d(i-1)}$ indicates its predecessor, and $t_\Delta = T_{d(i)} - T_{d(i-1)}$. By noting that t_{so} can fall anywhere in the range $T_{d(i-1)}$ to $T_{d(i)}$, the calibration error is bound by the following equation:

$$-t_\Delta/2 < t_{ce} \leq t_\Delta/2 \tag{3.8}$$

This relationship is illustrated in Figure 3.15, where the actual sampling offset of an arbiter is plotted versus its associated calibration error.

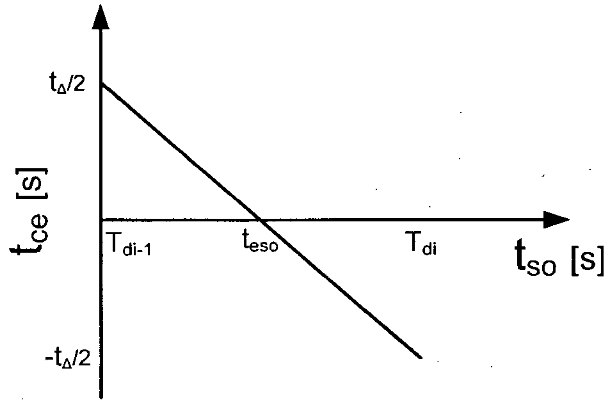


Figure 3.15: t_{so} versus t_{ce} for a noise-free arbiter using direct calibration.

Now that the limits of integration have been found, the probability density function of t_{ce} must be determined before integration can be performed. Since the goal of this exercise is to find the RMS calibration error of a noise-free arbiter for a given t_Δ, the arbiter's sampling offset must fall in the range $T_{d(i-1)}$ to $T_{d(i)}$ with equal probability, otherwise the results would be dependent on the actual value of the arbiter's sampling offset. In addition, the integral of this probability over the range $-t_\Delta/2$ to $t_\Delta/2$ must be equal to 1, since the arbiter has a fixed sampling offset that is greater than $T_{d(i-1)}$ but less than $T_{d(i)}$. These two conditions stipulate that for $-t_\Delta/2 < t_{ce} \leq t_\Delta/2$, $P(t_{ce}) = 1/t_\Delta$, as shown in Figure 3.16.

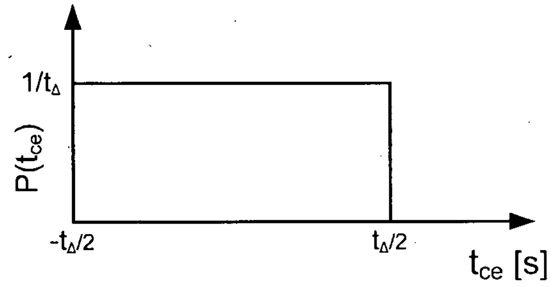


Figure 3.16: t_{ce} probability density function.

With this knowledge, the theoretical RMS t_{ce} of a noise-free arbiter calibrated using the direct calibration technique can be calculated as shown in Equation (3.9).

$$RMS(t_{ce}) = \sqrt{\frac{\int P(t_{ce})t_{ce}{}^2 dt_{ce}}{\int P(t_{ce})dt_{ce}}} = \sqrt{\int_{\frac{-t_\Delta}{2}}^{\frac{t_\Delta}{2}} P(t_{ce})t_{ce}{}^2 dt_{ce}} = \frac{t_\Delta}{2\sqrt{3}} \tag{3.9}$$

Equation (3.9) is plotted in Figure 3.17 along side the RMS calibration error obtained using the Matlab model described earlier, which incorporates the effects of thermal noise in an arbiter. It should be pointed out that the results of such a model can be highly dependent on the distribution of the arbiter sampling offsets. For example, if the sampling offsets of an array of arbiters fall in a very narrow range, one that is much smaller than the minimum t_Δ used during simulation, then the RMS calibration error may appear to be independent of σ_t. This result is intuitively wrong since the presence of thermal noise in an arbiter should result in a Gaussian-like CDF, from which a more accurate estimation of t_{so} can be made. To remove this dependency, t_Δ was fixed during simulation and σ_t was varied instead. The sampling offsets of the array of 100 arbiters were then assigned fixed values uniformly distributed over the range $T_{d(i-1)}$ to $T_{d(i)}$. This made for a fair comparison with the noise-free scenario.

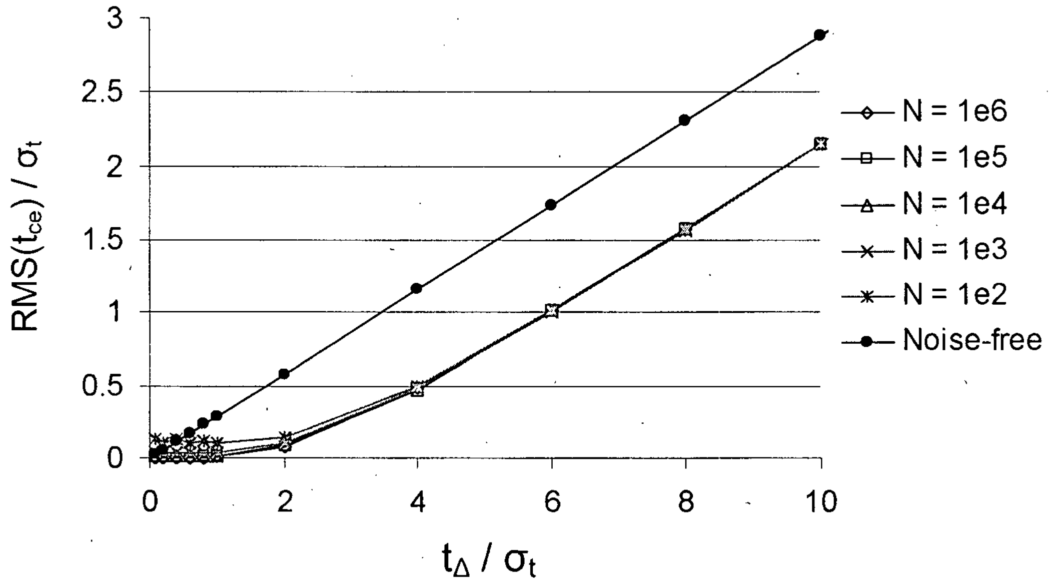


Figure 3.17: RMS t_{ce}/σ_t vs. t_Δ/σ_t using the direct calibration technique.

Inspection of Figure 3.17 reveals that for $2 \leq t_\Delta/\sigma_t \leq 10$, the presence of thermal noise in an arbiter significantly increases the accuracy of the arbiter sampling offset estimations obtained using the direct calibration technique. This result is expected since the presence of thermal noise in an arbiter contributes to a Gaussian-like CDF of the arbiter's sampling offset, from which a reasonably accurate estimation of t_{so} can be made. This is in contrast to the ramp-like CDF of a noise-free arbiter, for which the best approximation is a straight line interpolation, which has a significantly larger RMS error.

Inspection of Figure 3.17 also reveals that the gain in accuracy from the presence of thermal noise in an arbiter diminishes as t_Δ/σ_t is decreased from 2. Further insight in to this result can be acquired if Figure 3.17 is redrawn with logarithmic x and y axes, as shown in Figure 3.18.

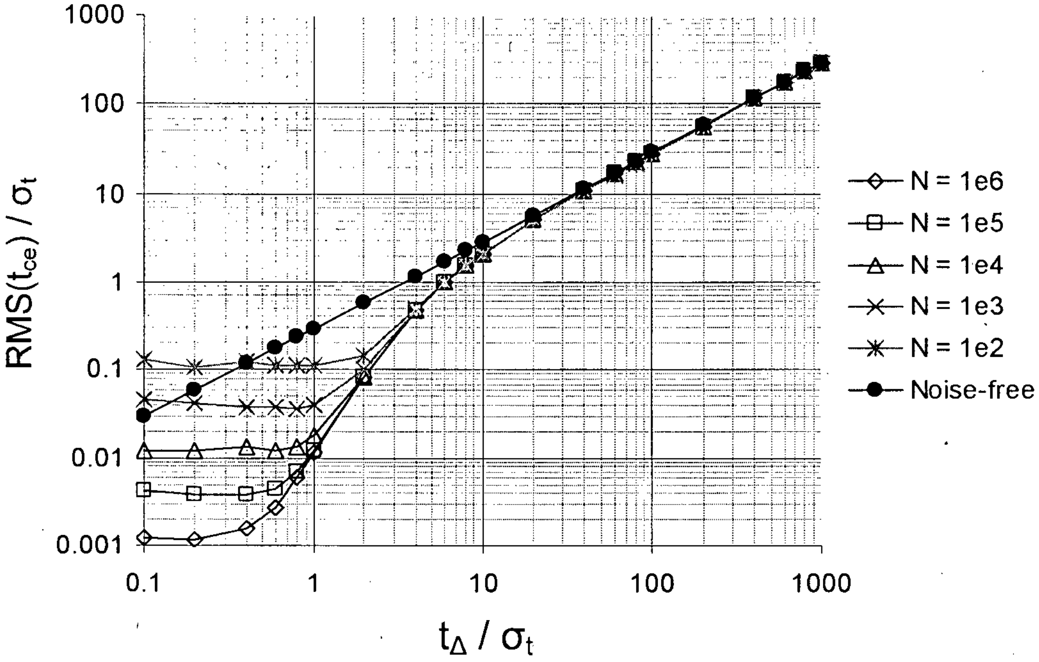


Figure 3.18: Log-log plot of RMS t_{ce}/σ_t vs. t_Δ/σ_t using the direct calibration technique.

From Figure 3.18, it can be observed that when t_Δ/σ_t is decreased, the RMS value of t_{ce}/σ_t reaches a saturation point somewhere in the range $0.2 \leq t_\Delta/\sigma_t \leq 1$, depending upon the number of repetitions performed. This implies that the actual RMS value of t_{ce} increases as one moves deeper into the saturation region, since σ_t increases as one moves closer to the y-axis, and the ratio of t_{ce} to σ_t is constant. By inspection of Figure 3.18, the rate at which the RMS value of t_{ce} increases as t_Δ/σ_t moves deeper into the saturation region is unclear. In order to clarify this, actual values of t_{ce} can be obtained if t_Δ is fixed to a particular value and σ_t varied. For example, the rate at which the RMS value of t_{ce} increases as t_Δ/σ_t decreases can be estimated from Figure 3.19, where t_Δ has been fixed at 10 ps and σ_t has been varied. The data required to plot Figure 3.19 is actually a special case of data displayed in Figure 3.18. For this reason Figure 3.18 is a more useful plot in a general sense, but not as convenient for a specific scenario.

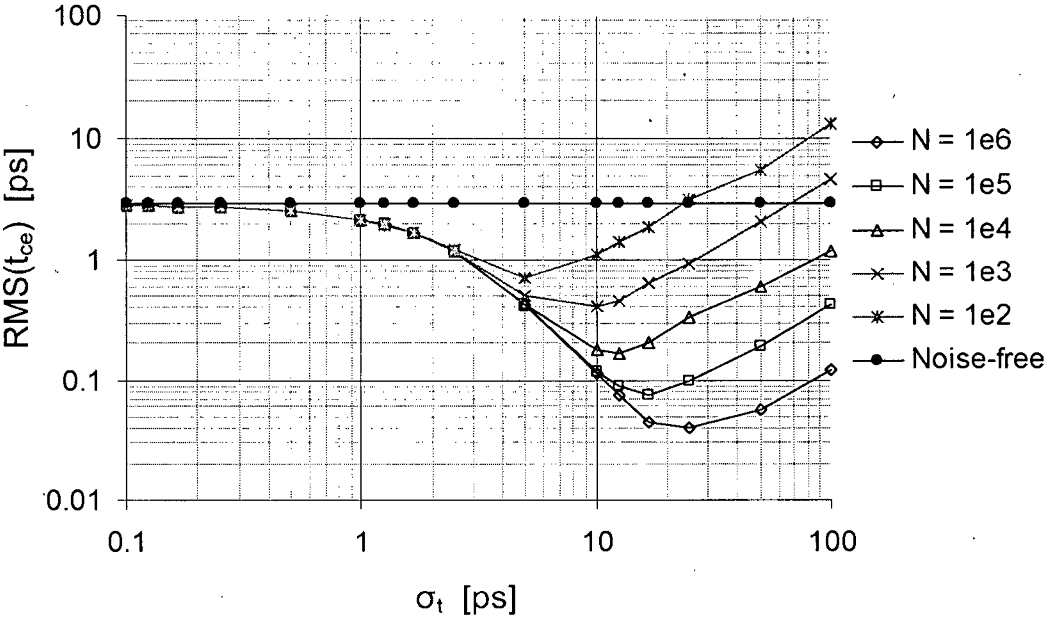


Figure 3.19: Log-log plot of RMS t_{ce} vs. σ_t when t_Δ = 10 ps, using the direct calibration technique.

Through inspection of Figure 3.19 it is apparent that an optimal ratio between t_Δ and σ_t exists, and is dependent upon the value of N. This ratio has been found to hold true for any value of t_Δ. Therefore, by determining the value of σ_t that minimizes t_{ce} for each of the five curves displayed in Figure 3.19, it is possible to determine the optimal ratio of t_Δ to σ_t given the desired number of repetitions. The optimal ratios for five different values of N are summarized in Table 3.1.

Table 3.1: Optimal ratio of t_Δ to σ_t given the number of repetitions performed.

N	t_Δ / σ_t
le2	2
le3	1
le4	0.8
le5	0.6
le6	0.4

This information can be used in the selection of t_Δ for the direct calibration technique. For example, if σ_t is equal to 0.35 ps [23] and N = 1e5, the value of t_Δ which would produce the lowest t_{ce} is 0.35 ps x 0.6 = 0.21 ps. Of course, the accurate generation of time intervals with such a temporal resolution is a very difficult task.

With this knowledge in hand, it is easier to explain why the gain in accuracy from the presence of thermal noise in an arbiter diminishes as t_Δ/σ_t is decreased from approximately 2. Through inspection of Figure 3.19, it can be seen that the farther one deviates from the optimal t_Δ/σ_t ratio, the larger the calibration error. For example, if σ_t is much smaller than its optimal value for a given t_Δ, hence much smaller than t_Δ, the difference between t_{ce} in this case and that of a noiseless arbiter calibrated using the same t_Δ becomes increasingly diminished. This result makes intuitive sense, since as σ_t becomes very small with respect to t_Δ, the CDF of the arbiter's sampling offset becomes less Gaussian-like and more ramp-like in appearance as the variation in the arbiter's sampling offset becomes less significant. Similarly, if σ_t is much larger than its optimal value for a given t_Δ, hence much larger than t_Δ, t_{ce} once again exceeds its minimum value. In this case the increased error can be attributed to the incorrect use of a polynomial curve fitting function on the relatively linear histogram that is produced.

3.2.2 Conclusions

In theory, the direct calibration technique can produce very accurate estimations of t_{so} given a sufficiently small t_Δ. In fact, the accuracy of this technique is limited only by t_Δ, which may be determined by the frequency difference of two oscillators. However in practice, this calibration technique has some very serious flaws. For example, any type of oscillator will have some amount of phase noise, and therefore will not have a perfectly stable frequency [32, 33]. Any instability in the frequency of either of the two oscillators can result in an increased error in the estimate of t_{so}. In fact, it may not even be possible to place a bound on the error as the amount of phase noise in either oscillator may be unknown.

Another problem with this calibration technique concerns the requirement that the sequence of T_ds must begin with a known T_d. One way to accomplish this may involve the use of an arbiter with a known sampling offset to detect alignment between the rising edge transitions of the two oscillators. However, as discussed in [23], the sampling offset of a reasonably sized arbiter may vary from its intended value by as much as 25 picoseconds. An error of 25 picoseconds in the initial T_d will propagate to the estimate of t_{so} for each arbiter. One way to alleviate this problem is to oversize the transistors in the arbiter that is used for alignment. This will help to reduce the arbiter's sensitivity to process variations. However, even if an arbiter with a known sampling offset is used to detect alignment between the rising edge transitions of the oscillators, there is still a quantization error in the edge alignment of at most t_Δ seconds due to the finite difference in the frequencies of the two oscillators. Also, any mismatch in the *START* and *STOP* signal paths will introduce some skew between them, and this skew will alter the sampling offset of the alignment arbiter by an unknown amount, adding another error to the estimate of t_{so}.

While the direct calibration technique is conceptually rather simple, it is not used in practice due to its many shortcomings; the most severe being the restriction placed on t_Δ. The accurate generation of known time intervals with picosecond resolution is very difficult to achieve on-chip, and therefore renders this calibration technique ineffective for embedded applications.

3.3 Relative Offset Calibration Technique

A technique capable of determining the relative sampling offsets of an array of arbiters is presented in [23, 25]. This technique analyzes the "bubbles" in the output codeword of an SOTDC. A codeword is said to be "bubble-free" if there is at most one location in the codeword where adjacent bits differ. For example, an 8-bit "bubble-free" codeword may look like the following: "00001111". A codeword is said to contain a "bubble" if there are three locations where adjacent bits differ, as shown in the following codeword: "00101111".

If there are more than three locations in a codeword where adjacent bits differ, then the codeword is said to contain more than one "bubble".

"Bubbles" may appear in an SOTDC codeword when its resolution is comparable to σ_t. For example, if arbiters A_1 and A_2 have sampling offsets of 100 ps and 101 ps, respectively, and a 100.5 ps time interval is applied to the inputs of both arbiters, the most probable outcome is that A_1 will output a logic '1' and A_2 will output a logic '0'. However, there is a significant probability of the reverse scenario occurring, i.e., A_1 outputs a logic '0' and A_2 output a logic '1'. If this experiment is performed a sufficient number of times, this counter intuitive outcome is inevitable, and will occur with a certain probability. The ratio of these two probabilities can be used to determine the difference in the sampling offsets of arbiters A_1 and A_2. For example, if the probability of the more likely outcome is denoted as $P_{A1A2}(10)$, i.e., the output of A_1 is a logic '1' and the output of A_2 is a logic '0', and the probability of the less likely outcome is denoted as $P_{A1A2}(01)$, then the ratio of these two probabilities, $r = P_{A1A2}(01)/P_{A1A2}(10)$, depends only on δ, which is the ratio of the difference in the sampling offsets of the two arbiters to $2\sigma_t$, i.e., $\delta = (t_{s\alpha A2} - t_{s\alpha A1})/2\sigma_t$. The exact relationship between these two ratios is derived in [23], where the following equation is produced:

$$r \equiv \frac{P_{A1A2}(01)}{P_{A1A2}(10)} = \frac{1+\sqrt{\pi}\delta(erfcx(-\delta))}{1-\sqrt{\pi}\delta(erfcx(\delta))} \tag{3.10}$$

The "*erfcx*" terms in the right hand side of the preceding equation are instances of the scaled complementary error function.

In summary, the author in [23] proposes measuring r and inverting Equation (3.10) in order to find the relative sampling offsets of a pair of arbiters in terms of σ_t. However, the author does not present a viable on-chip solution for obtaining σ_t. In addition, a critical assumption about the mean sampling offset of the arbiters is made by the author, and is stated in [25]. In this work the author states that if the absolute sampling offsets of the arbiters are to be determined, then the mean sampling offset of the arbiters must be known. The author suggests that the mean sampling offset of a large number of arbiters can be predicted if the sampling offsets of the arbiters are altered by process variation alone, i.e., no attempt is made during the design of the arbiters to differentiate their sampling offsets from one another. In such a case the author predicts that the sampling offsets of the arbiters would follow a Gaussian distribution, as shown in Figure 3.20, where the mean sampling offset, denoted as μ_{so}, is equal to the intended sampling offset of the arbiters.

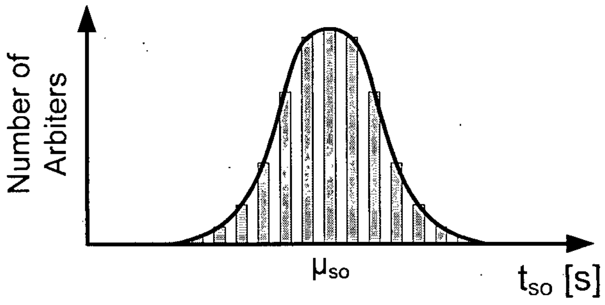


Figure 3.20: Gaussian distribution of arbiter sampling offsets due to process variation.

For example, if an array of 64 arbiters, designed to be perfectly symmetric, is fabricated on a single die, then the author predicts that the actual sampling offsets of the arbiters will follow a Gaussian distribution with a mean of zero.

Several problems exist with this assumption. Firstly, since an SOTDC consists of a finite number of arbiters, it is difficult to ensure that the sampling offsets will vary according to a Gaussian distribution. While it is true that a distribution which closely matches a Gaussian may be obtainable if an SOTDC is constructed using a very large number of arbiters, perhaps greater than 1000, the penalty to be paid in such a case is an excessive use of silicon area. Also, each arbiter may be subject to some constant amount of process variation which results in a common shift in the sampling offsets of all the arbiters. Such a scenario is not accounted for in the preceding assumption and will therefore increase the error in the estimations of the arbiter sampling offsets.

If the mean sampling offset of an array of arbiters cannot be determined with a reasonable degree of confidence, then for the purpose of time interval measurement, the only useful information that can be extracted from the arbiters is the amount of variation in a series of time intervals. For example, the standard deviation of a series of time intervals could be measured, however not the mean. Due to the aforementioned issues, the relative offset calibration technique is more interesting from a theoretical perspective than a practical one.

3.4 Added Noise Calibration Technique

A calibration technique based on "added noise" has been described in [26]. This technique is fundamentally identical to the direct calibration technique, with the exception of one important modification. Since σ_t has been measured to be approximately 0.35 ps [23], the direct calibration technique requires the accurate generation of known time intervals with picosecond temporal resolution. This is a very difficult task to achieve on-chip. To better illustrate this requirement, ten different values of t_Δ have been simulated using the Matlab model described in section 3.2.1 with N = 100 000, while σ_t has been varied. The results of these simulations are shown in Figure 3.21.

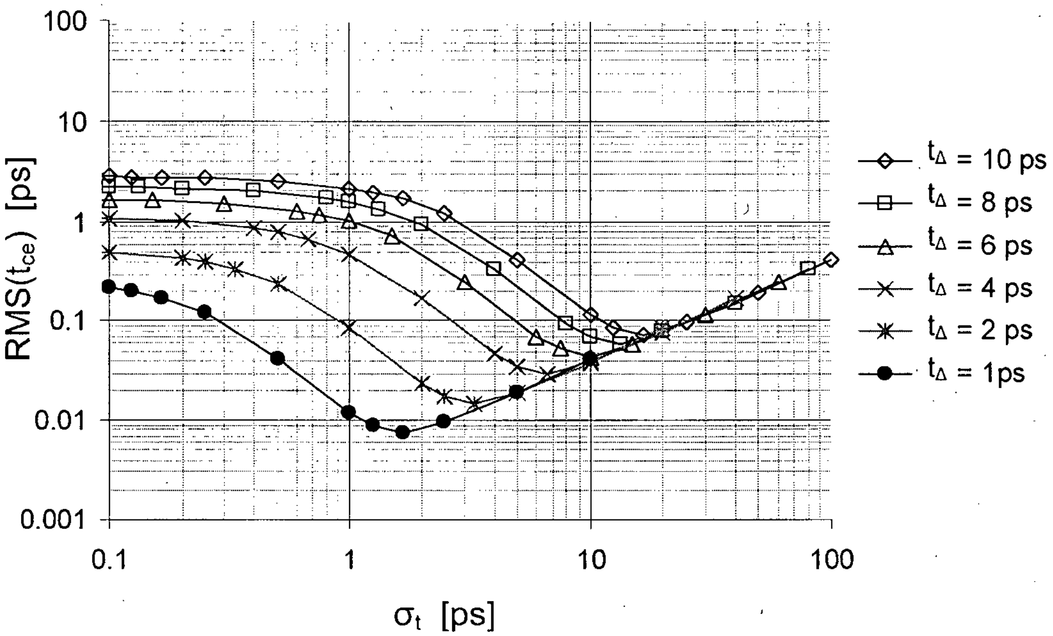


Figure 3.21: Log-log plot of RMS t_{ce} vs. σ_t when N = 100 000, using the direct calibration technique.

From Figure 3.21 it can be seen that if σ_t = 0.35 ps, t_Δ cannot be greater than 4 ps if t_{ce} is to be kept below 1 ps. In reality, calibration accuracies greater than 1 ps are required for high-resolution SOTDCs. While it is possible to increase N in order to alleviate some of the restrictions placed on t_Δ, demands placed on the total calibration time usually limit N to 100 000 or less [34].

In order to circumvent the restrictions placed on t_Δ, the authors in [26] suggest adding Gaussian temporal noise to the arbiters in an SOTDC. In fact, the authors advocate adding Gaussian temporal noise with a standard deviation much larger than σ_t. A large amount of Gaussian temporal noise drastically alters the restrictions placed on the temporal resolution of the time intervals. For example, through inspection of Figure 3.21 it can be seen that if t_Δ = 1ps and σ_t = 0.35 ps, the predicted t_{ce} is 0.07 ps. Now if σ_t is increased to 17 ps, t_Δ can be increased to 10 ps while still maintaining the same t_{ce}. That is, a 49-fold increase in σ_t allows for a 10-fold increase in t_Δ without an increase in N or t_{ce}.

In order to understand how the authors in [26] propose to add Gaussian temporal noise to the arbiters in an SOTDC, a time domain model of thermal and added noise in a biased arbiter must first be presented, as shown in Figure 3.22.

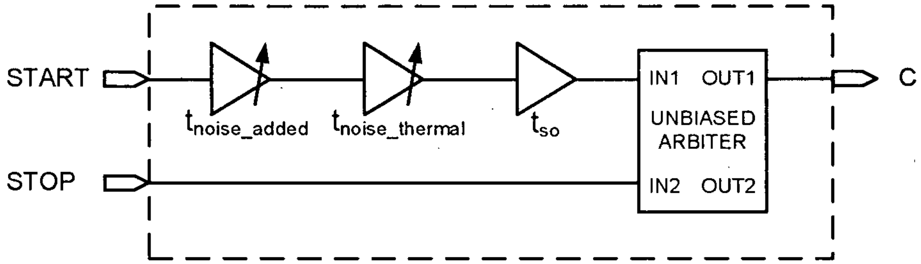


Figure 3.22: Time domain model of added and thermal noise in a biased arbiter.

The added Gaussian noise is modelled with the inclusion of a second variable delay buffer. Both the added Gaussian noise and the intrinsic thermal noise act to vary the sampling offset of the arbiter, however to different extents. An illustration of the contribution of each noise source, superimposed on one another, is shown in Figure 3.23.

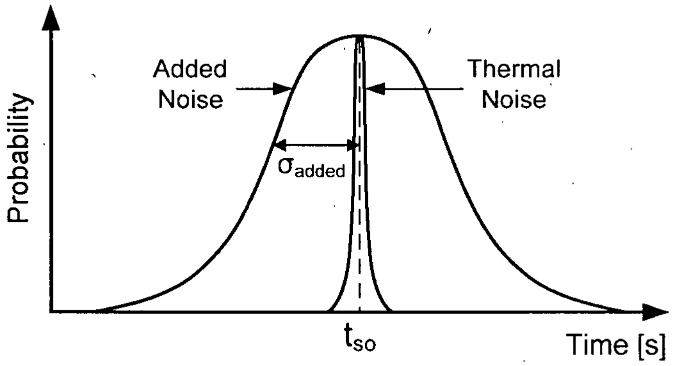


Figure 3.23: Arbiter sampling offset PDF with thermal and added noise.

Assuming the noise sources are independent, the standard deviation of the arbiter's sampling offset can be determined with the aid of the following equation:

$$\sigma_{total} = \sqrt{\sigma_t^2 + \sigma_{added}^2} \tag{3.11}$$

If the standard deviation of the added noise is chosen to be much greater than that of the thermal noise, i.e., $\sigma_{added} >> \sigma_t$, then σ_{total} can be accurately approximated as σ_{added}.

Instead of injecting Gaussian temporal noise directly into the arbiters themselves, the authors in [26] suggest modulating the time intervals. This clever idea provides a simple mechanism to effectively vary the sampling offset of an arbiter according to a Gaussian distribution without the need to actually change the arbiter's circuitry. Figure 3.24 illustrates how a CDF of an arbiter's sampling offset is created from a sequence of time intervals, where each time interval is distributed according to a Gaussian distribution with a standard deviation of σ_{added}.

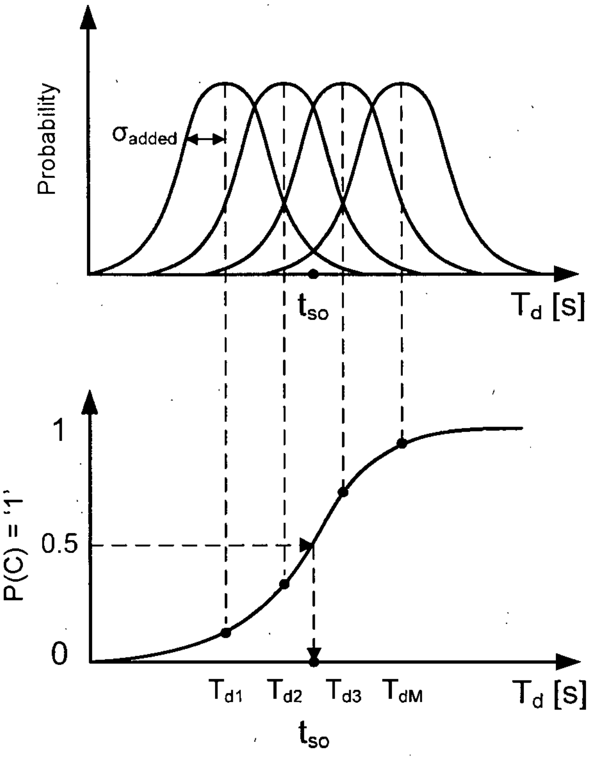


Figure 3.24: Addition of Gaussian temporal noise to a sequence of time intervals in order to create an arbiter sampling offset CDF.

In order to generate a sequence of accurately known time intervals with Gaussian distributions, the authors in [26] propose the use of a configuration as illustrated in Figure

3.25. In this configuration, a production tester or an on-chip DLL is used to generate the accurately known time intervals. The time intervals are then modulated by the Gaussian control voltage of a variable delay buffer.

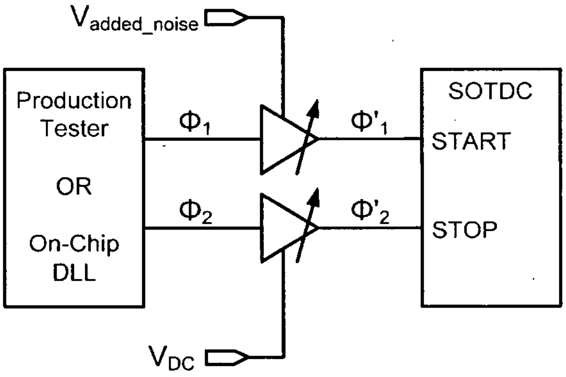


Figure 3.25: Added noise calibration technique implementation [26].

The feasibility of such an approach will be discussed in section 3.4.2.

3.4.1 Analysis

The authors in [26] have created a Matlab model of the added noise-based calibration technique and have reported the results of a small number of simulations. These results are displayed in Table 3.2.

Table 3.2: Reported results from Matlab simulation of the added noise-based calibration technique (t_Δ = 40 ps, σ_t = 250 ps, N = 100 000) [26].

Arbiter Offset [ps]	Calibrated Offset [ps]	Error [ps]
-35.00	-34.60	0.40
-18.31	-17.91	0.40
-2.00	-2.12	-0.12
5.00	5.30	0.30
6.00	5.43	-0.57
17.40	17.10	-0.30
27.50	27.47	-0.03

The authors in [26] have chosen to perform their simulations with t_Δ = 40 ps, σ_t = 250 ps, and N = 100 000. Calculation of the RMS error of the results in Table 3.2 yields 0.35 ps. This result can be compared with the predictions of the Matlab model of the direct calibration technique described in section 3.2.1. It should be noted that such a comparison is valid as the added noise calibration technique is theoretically identical to the direct calibration technique. The only difference between the two techniques is the amount of Gaussian noise in an arbiter.

Using the Matlab model of the direct calibration technique described in section 3.2.1, it is possible to plot the RMS t_{ce} versus σ_t when t_Δ = 40 ps. Such a plot is shown in Figure 3.26.

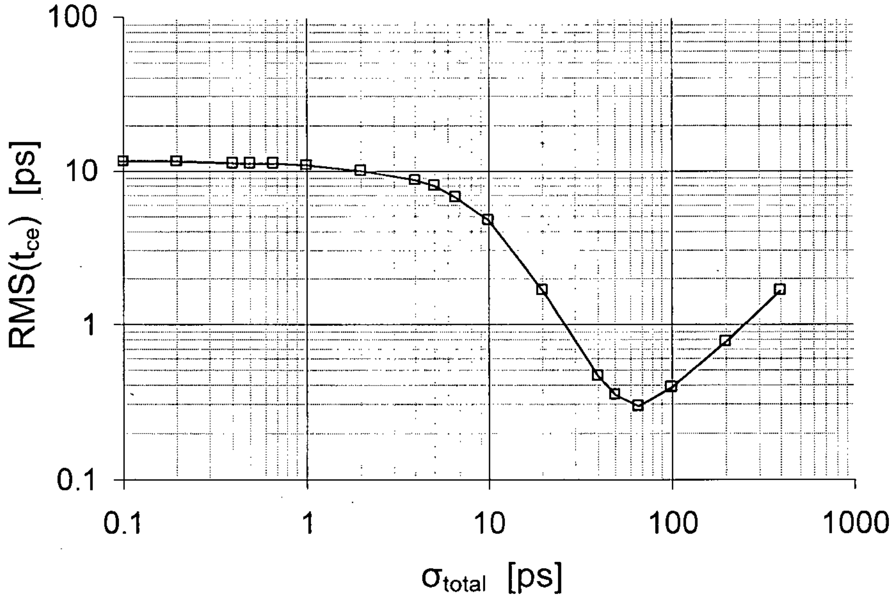


Figure 3.26: Log-log plot of RMS t_{ce} vs. σ_t when t_Δ = 40 ps and N = 100 000, using the model of the direct calibration technique described in section 3.2.1.

Inspection of Figure 3.26 reveals that according to the Matlab model described in section 3.2.1, the RMS value of t_{ce} is approximately 1 ps if σ_t = 250 ps. However, the authors in [26] present data with an RMS value of 0.35 ps. This discrepancy may be due to the fact that a relatively small number of arbiters (7) have been simulated in [26], whereas 100 arbiters have been simulated with the Matlab model described in section 3.2.1. Further inspection of

Figure 3.26 reveals that according to the Matlab model described in section 3.2.1, it is possible to obtain an RMS value of t_{ce} as low as 0.3 ps when N = 100 000. However, in order to do so, σ_t must be reduced to 67 ps. This value agrees with the optimal t_Δ to σ_t ratio of 0.6 when N = 100 000, as shown in Table 3.1. Calculation of the ratio of t_Δ to σ_t used in [26] produces a result of 0.16, indicating that σ_t should be decreased in order to reduce the RMS value of t_{ce}. While it may be true that $\sigma_t = 250$ ps produces a relatively low RMS t_{ce} for the arbiter sampling offsets specified in Table 3.2, this result may not hold true for a more general distribution of arbiter sampling offsets, such as the uniform distribution used in the Matlab model described in section 3.2.1.

The amount of time required by the added noise-based calibration technique to perform calibration is proportional to N and M, the number of repetitions and the number of time intervals, respectively, and inversely proportional to *f*, the frequency at which the time intervals are applied to the SOTDC *START* and *STOP* signals. This relationship is summarized with the following equation [26].

$$t_{cal} = \frac{MN}{f} \tag{3.12}$$

This is the same amount of time required by the direct calibration technique. However, some additional time is required to apply a curve fitting algorithm to the histogram data.

3.4.2 Conclusions

The added noise-based calibration technique proposed in [26] is useful in the sense that it allows the step size of the time intervals (t_Δ) to be increased while still maintaining the same level of calibration accuracy. However, this method does not alleviate the need for accurately known time intervals. This is a significant issue as it infers that either the external production tester or the on-chip DLL must generate known time intervals with picosecond accuracy. As the accuracy of the time intervals suffers, so does the accuracy of the calibration results. For example, if all of the time intervals generated by the external tester or the on-chip DLL are 1 ps greater than their assumed values, then the RMS value of the calibration error will increase by 1 ps. This error will propagate to the results of time interval measurements made by the SOTDC.

In addition, the on-chip variable delay buffers must be calibrated in order to ensure that they do not add unwanted skew between Φ_1 and Φ_2, and also to ensure that a linear relationship exists between the voltage of the control signals and the delay of the buffers. For these reasons it can be said that the method of implementation of the added noise-based calibration technique proposed in [26] is neither an ideal nor a complete solution. The SOTDC calibration technique proposed in Chapter 4 does not require knowledge of the time intervals used for calibration.

Chapter 4

Proposed SOTDC Calibration Technique

As discussed in Chapter 3, several SOTDC calibration techniques exist. However, it has been shown that all such techniques suffer from at least one serious limitation, thus rendering these proposals either unfeasible or insufficiently accurate. In order to address the need for a feasible and accurate SOTDC calibration technique, a new calibration technique has been developed. This technique leverages some of the advantages of the added noise-based calibration technique, while omitting some of its limitations.

4.1 Simplified Proposed Calibration Technique

The calibration technique proposed in this thesis relies upon the availability of two oscillators with a known frequency difference. That is, two oscillators, namely *oscA* and *oscB*, are required. If the frequency of *oscA* and *oscB* are denoted as f_A and f_B, respectively, then the frequency difference of the two oscillators can be denoted as f_Δ.

$$f_\Delta = f_B - f_A \tag{4.1}$$

The same analysis can be performed in the time domain if the period of *oscA* and *oscB* are denoted as T_A and T_B, respectively, and the period difference of the two oscillators is denoted as T_Δ.

$$T_\Delta = T_B - T_A \tag{4.2}$$

When *oscA* and *oscB* oscillate freely, the difference in time between the rising edge transitions of each oscillator can be interpreted as a sequence of time intervals (T_ds), with each time interval being T_Δ seconds shorter or longer than its predecessor, as shown in Figure 4.1.

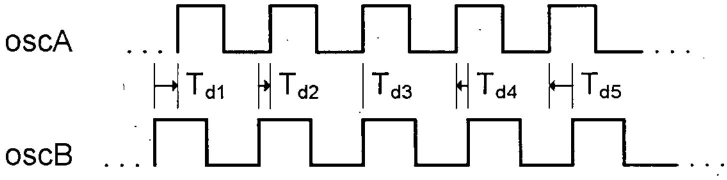


Figure 4.1: Time intervals created by two free-running oscillators.

Figure 4.1 depicts the relative temporal locations of the rising and falling edge transitions of *oscA* and *oscB*. These waveforms are drawn with the assumption that both oscillators are perfectly stable, and $f_B < f_A$, or equivalently, $T_B > T_A$. A perfectly stable oscillator is defined as an oscillator that has a constant frequency, and hence a constant period. For the remainder of this section, oscillators will be assumed to be perfectly stable. In addition, arbiters will be assumed to be noise-free. Therefore, it will be assumed that the sampling offset of an arbiter is a constant value, independent of time. This assumption will make the explanation of the simplified proposed calibration technique easier to follow. The presence of temporal noise in the sampling offset of an arbiter will be considered in section 4.2.

If the instant in time at which the output of *oscB* produces a rising edge transition is denoted as t_{oscB}, and if t_{oscA} is defined analogously for *oscA*, then each time interval can be expressed mathematically as:

$$T_{d(i)} = t_{oscB(i)} - t_{oscA(i)} \tag{4.3}$$

where i denotes the i^{th} rising edge transition of an oscillator. Following this definition of a time interval, a sequence of time intervals N elements long, i.e., $T_{d1} \dots T_{dN}$, generated from the output of *oscA* and *oscB* can be defined as shown in Equation (4.4).

$$\begin{aligned} T_{d(i)} &= \{T_{d1}, T_{d2}, T_{d3}, T_{d4}, \dots T_{dN-1}, T_{dN}\}; \quad 1 \le i \le \mathrm{N} \\ &= \{T_{d1}, T_{d1} + T_{\Delta}, T_{d1} + 2T_{\Delta}, T_{d1} + 3T_{\Delta}, \dots T_{d1} + (\mathrm{N}-1)T_{\Delta}\}; \quad 1 \le i \le \mathrm{N} \end{aligned} \tag{4.4}$$

Through inspection of Equation (4.4) it can be seen that a general formula exists for the duration of a time interval generated from the output of two oscillators of different frequency. Such a formula is written in Equation (4.5), where $i > 0$.

$$T_{d(i)} = T_{d1} + (i-1)T_{\Delta} \tag{4.5}$$

Equation (4.5) describes a linearly increasing sequence of time intervals, as shown in Figure 4.2.

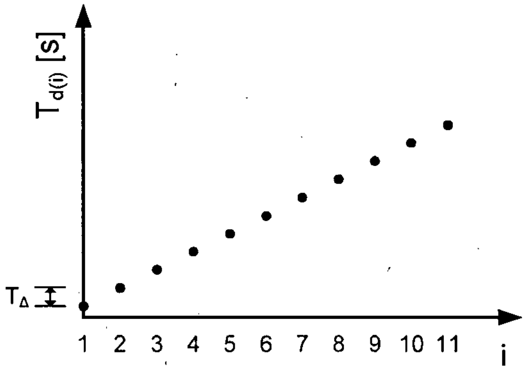


Figure 4.2: Sequence of linearly increasing time intervals.

As discussed in Chapter 3, a periodic sequence of known time intervals can be used to determine the sampling offsets of an array of arbiters. It is possible to generate a periodic sequence of unknown time intervals from the output of two oscillators of different frequency if $T_{d(i)}$ is restricted to a finite interval. For example, if $T_{d(i)}$ is bound by the following equation,

$$0 < T_{d(i)} \leq T_A \qquad (4.6)$$

then a periodic sequence of time intervals can be generated if T_A/T_Δ is an integer. Figure 4.3 illustrates how a periodic sequence of time intervals can be produced from the output of two oscillators of different frequency, where $T_A/T_\Delta = 5$.

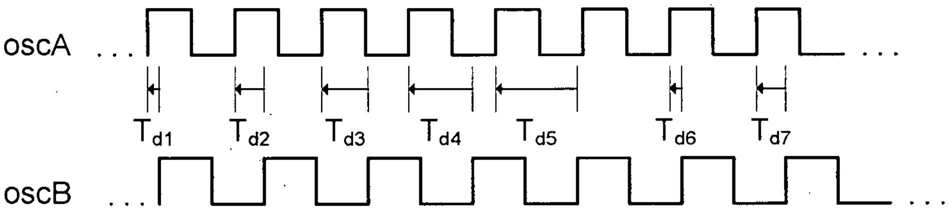


Figure 4.3: Periodic time intervals created by two free-running oscillators.

Inspection of Figure 4.3 reveals that $T_{d6} = T_{d1}$ and $T_{d7} = T_{d2}$, or more generally, $T_{d(i)} = T_{d(i-5)}$ for $i > 5$. Therefore in general, if T_A/T_Δ is an integer, then a periodic sequence of time intervals with T_A/T_Δ unique values may be created from the output of two oscillators of different frequency. There is no need to physically impose a limit on the size of $T_{d(i)}$. This limit naturally occurs if one oscillator is used as a reference edge generator, and the relative temporal location of the other is used to indicate the duration of the time interval, as is illustrated in Figure 4.3. Equation (4.5) can be rewritten to account for the periodic nature of the time intervals generated from the output of two oscillators of different frequency, assuming T_A/T_Δ is an integer.

$$T_{d(i)} = T_{d1} + [(i - 1) \bmod (T_A/T_\Delta)]\, T_\Delta; \quad i > 0 \qquad (4.7)$$

The periodic sequence of time intervals described by Equation (4.7) is plotted in Figure 4.4, assuming $T_A/T_\Delta = 5$.

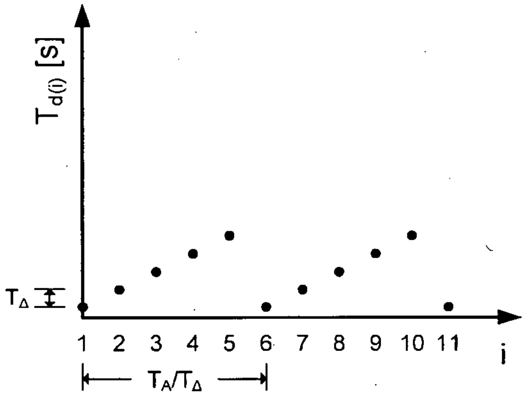


Figure 4.4: Periodic sequence of time intervals generated from the output of *oscA* and *oscB* assuming $T_A/T_\Delta = 5$.

Inspection of Equation (4.7) reveals that one of the time intervals generated from the output of two oscillators of different frequency must be accurately known before the entire sequence of time intervals can be predicted. However, without this information, the entire sequence of time intervals is unknown, and therefore of no use to any of the calibration techniques discussed in Chapter 3. Their values could be determined if an arbiter with a known sampling offset is used to detect alignment between the two oscillators. However, if it was possible to determine the sampling offset of an arbiter, then a calibration technique would not be required in the first place. Even if such an arbiter was available, any differences in the routing of the arbiter's inputs could significantly alter the arbiter's sampling offset. This in turn would affect the predicted values of the sequence of time intervals, and thus the accuracy of the calibration technique. As a result, the only useful information that can be directly extracted from a periodic sequence of unknown time intervals is the temporal difference between arbiter sampling offsets. This can be accomplished by counting the number of oscillator cycles elapsed between the "switching-events" of two arbiters, as illustrated in Figure 4.5.

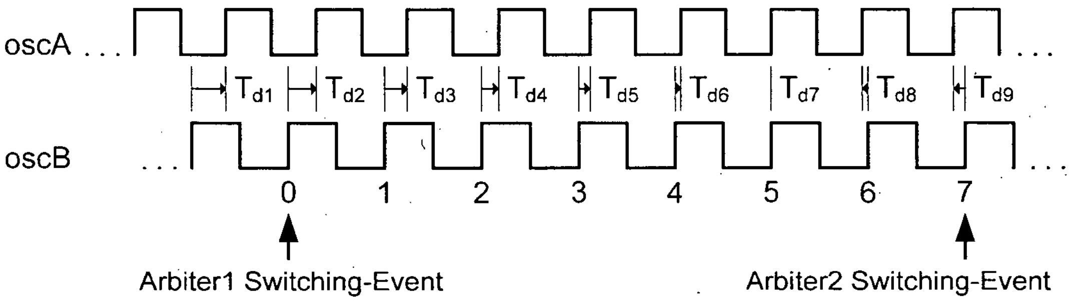


Figure 4.5: Determining the relative sampling offsets of two arbiters.

In order to explain the meaning of an arbiter's "switching-event", an arbiter's response to two important time intervals must be understood. As discussed in Chapter 2, when a time interval which is less than the sampling offset of an arbiter (t_{so}) is applied to its inputs, the arbiter responds by asserting its *OUT2* output while maintaining a low logic level on its *OUT1* output, i.e., *OUT1* = '0' and *OUT2* = '1'. On the other hand, if a time interval which is greater than or equal to the sampling offset of an arbiter is applied to its inputs, the arbiter responds by asserting its *OUT1* output while maintaining a low logic level on its *OUT2* output, i.e., *OUT1* = '1' and *OUT2* = '0'. Therefore, if a sequence of time intervals is applied to the inputs of an arbiter, such as the one plotted in Figure 4.4, where $T_{d1} < t_{so}$, and $T_{dQ} \geq t_{so}$ ($Q = T_A/T_\Delta$ is an integer), then for some $T_{d1} < T_{d(i)} \leq T_{dQ}$, the arbiter's response will change from *OUT1* = '0' and *OUT2* = '1' to *OUT1* = '1' and *OUT2* = '0'. This event is known as the arbiter's "switching-event", and signifies that the arbiter's sampling offset has been surpassed or equalled by the most recently applied time interval, $T_{d(i)}$. Detecting the switching-event of an arbiter is useful as the sampling offset of the arbiter can then be estimated as $T_{d(i-1)} < t_{so} \leq T_{d(i)}$, where both $T_{d(i-1)}$ and $T_{d(i)}$ are unknown.

Inspection of Figure 4.5 reveals that the switching-event of *Arbiter2* occurs seven cycles after the switching-event of *Arbiter1*. From this information one might surmise that the sampling offset of *Arbiter2* is greater than that of *Arbiter1* by $7T_\Delta$ seconds, or equivalently, $t_{so2} = t_{so1} + 7T_\Delta$. However, it might be incorrect to form this assumption. In order to illustrate this point, Figure 4.6 may be of use. Inspection of Figure 4.6 reveals that the difference in the sampling offsets of *Arbiter1* and *Arbiter2* may vary from nearly $6T_\Delta$ seconds to almost $8T_\Delta$ seconds, while still maintaining an oscillator cycle count of seven.

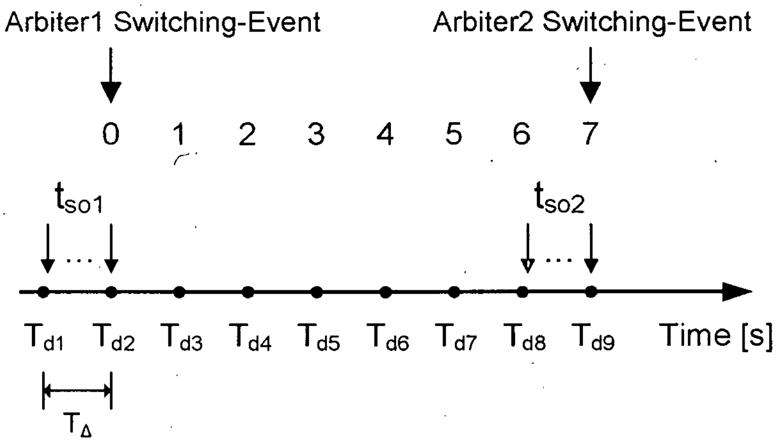


Figure 4.6: Variation in arbiter sampling offsets while still maintaining a constant cycle count.

Therefore, the number of oscillator cycles elapsed between the switching-events of two arbiters provides enough information to estimate the relative temporal spacing between the sampling offsets of the arbiters to within a range of $2T_\Delta$ seconds. Using the scenario illustrated in Figure 4.5 as an example, Equation (4.8) describes the sampling offset of *Arbiter2* in terms of the sampling offset of *Arbiter1*.

$$t_{so1} + 6T_\Delta \leq t_{so2} \leq t_{so1} + 8T_\Delta \tag{4.8}$$

Until this point, only perfectly symmetric or positively biased arbiters have been discussed. It is also possible to construct a negatively biased arbiter. The behaviour of a negatively biased arbiter is illustrated in Figure 4.7. A buffer delay (τ_{del}) has been inserted before input *IN2* of a perfectly symmetric arbiter in order to mimic the behaviour of a negatively biased arbiter. However in reality, the sizes of transistors within the arbiter are usually altered in order to induce a bias.

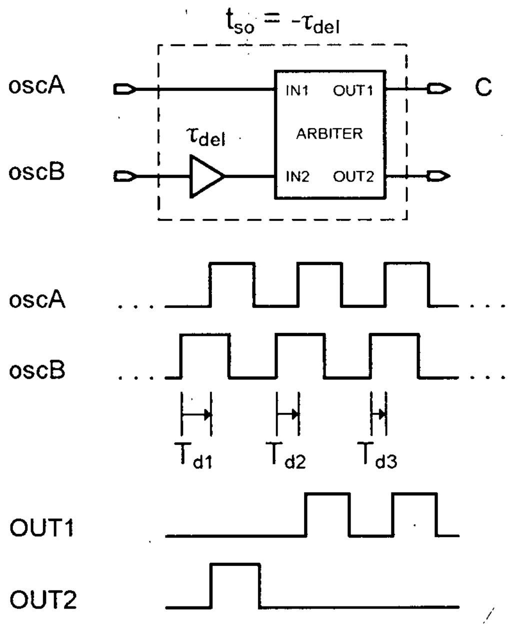


Figure 4.7: Behaviour of a negatively biased arbiter.

Inspection of Figure 4.7 reveals that the sampling offset of the illustrated arbiter is greater than T_{d1} but less than or equal to T_{d2}. Since $T_{d(i)} = t_{oscB(i)} - t_{oscA(i)}$, both T_{d1} and T_{d2} are negative. As the difference between T_{d1} and T_{d2} is made increasingly small, the sampling offset of the arbiter can be found to be equal to $-\tau_{del}$, as expected.

Returning to the subject of Figure 4.5, it can be seen that the sampling offset of *Arbiter1* is a negative number. In addition, it can be seen that the magnitude of t_{so1} is greater than that of t_{so2}, which is a positive number. Now, if the inputs to *Arbiter2* are somehow reversed, and its switching-event is redefined to occur when its input response changes from *OUT1* = '1' and *OUT2* = '0' to *OUT1* = '0' and *OUT2* = '1', then the arbiter's sampling offset changes sign but not magnitude, as illustrated in Figure 4.8. This can be attributed to the topology of a symmetric CMOS arbiter. A typical CMOS D flip-flop does not share this property as its setup time is dependent upon the logic values of its present and previous inputs.

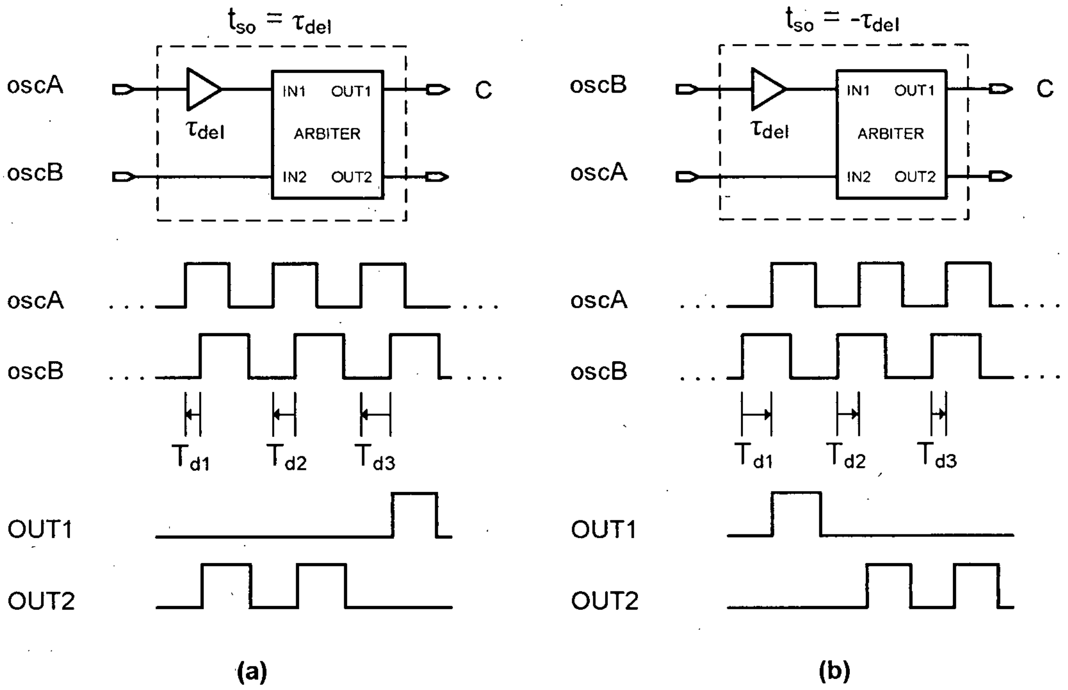


Figure 4.8: Behaviour of a positively biased arbiter (a), and a positively biased arbiter with reversed inputs (b).

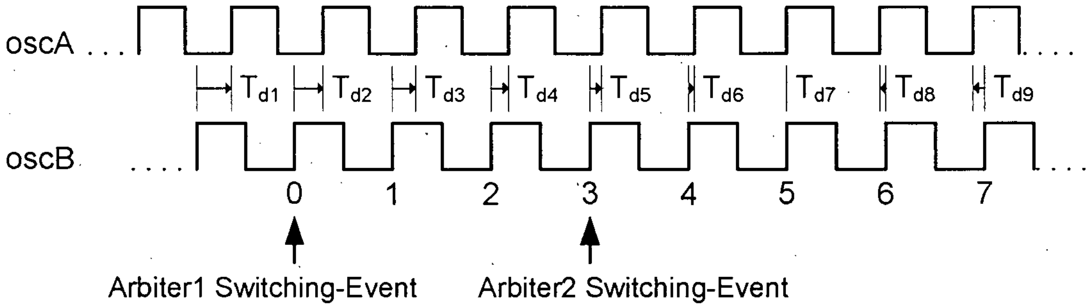
This change in sign is extremely useful, as it provides a method of obtaining a second piece of information regarding the relationship between the sampling offsets of *Arbiter1* and *Arbiter2*. For example, if the inputs to *Arbiter2* are reversed while the inputs to *Arbiter1* remain unchanged, the relative temporal difference between the switching-events of the two arbiters changes. In fact, it decreases by exactly $2t_{so2}$. Therefore in theory, the sampling offset of *Arbiter2* can be estimated if two pieces of information concerning the relative temporal difference between the switching-events of *Arbiter1* and *Arbiter2* are obtained. The first piece of required information is the number of oscillator cycles elapsed between the switching-events of the two arbiters, with the inputs to both arbiters as illustrated in Figure 4.8 (a). The second piece of required information is the number of oscillator cycles elapsed between the switching-events of the two arbiters when the inputs to *Arbiter2* are reversed, as illustrated in Figure 4.8 (b). From this information, the sampling offset of *Arbiter2* can be estimated using Equation (4.9).

$$2t_{so2} = \left(cycleCount_{Arbiter2\_normal\_inputs} \cdot T_{\Delta}\right) - \left(cycleCount_{Arbiter2\_reversed\_inputs} \cdot T_{\Delta}\right)$$

$$t_{so2} = \left(cycleCount_{Arbiter2\_normal\_inputs} - cycleCount_{Arbiter2\_reversed\_inputs}\right)\frac{T_{\Delta}}{2} \qquad (4.9)$$

This result is quite powerful as it demonstrates that it is possible to estimate the sampling offset of an arbiter using two oscillators with a known frequency difference, a counter circuit, and a second arbiter with an unknown sampling offset. The need to generate a sequence of known time intervals has been eliminated.

In order to determine the theoretical accuracy of Equation (4.9), it would be helpful if the example illustrated in Figure 4.5 included information concerning the behaviour of *Arbiter2* when its inputs are reversed. However, this would require knowledge of the sampling offset of *Arbiter2*. Therefore, in order to deduce the accuracy of Equation (4.9), a value must be chosen for t_{so2}. If t_{so2} is arbitrarily fixed at $2T_{\Delta}$, then the number of oscillator cycles elapsed between the switching-events of *Arbiter1* and *Arbiter2*, when the inputs to *Arbiter2* are reversed, can be predicted. Under normal circumstances, i.e., when the inputs to *Arbiter2* are not reversed, seven oscillator cycles are elapsed between the switching-events of *Arbiter1* and *Arbiter2*. However, when the inputs to *Arbiter2* are reversed, its sampling offset changes sign, and is therefore equal to $-2T_{\Delta}$. Since each time interval differs by T_{Δ} seconds from its predecessor, a decrease in t_{so2} of $4T_{\Delta}$ seconds corresponds to a decrease of 4 oscillator cycles. Therefore, 3 oscillator cycles are elapsed between the switching-events of *Arbiter1* and *Arbiter2* when the inputs to *Arbiter2* are reversed, as illustrated in Figure 4.9.



T_{d1} T_{d2} T_{d3} T_{d4} T_{d5} T_{d6} T_{d7} T_{d8} T_{d9}

Figure 4.9: Oscillator cycle count when the inputs to *Arbiter2* are reversed.

While it is important to know the number of oscillator cycles elapsed between the switching-events of *Arbiter1* and *Arbiter2*, for the purposes of determining the accuracy of Equation (4.9), it is more insightful to know the temporal range of t_{so2} that produces an oscillator cycle count of three. A simple modification of Equation (4.8), which provides an estimate of t_{so2} in terms of t_{so1} assuming the inputs to *Arbiter2* are not reversed, can yield such results. Since it is known that t_{so2} with reversed inputs is $4T_\Delta$ seconds less than t_{so2} without reversed inputs, $4T_\Delta$ seconds can be subtracted from all of the terms in Equation (4.8) to make the required modification, as shown below.

$$t_{so1} + 6T_\Delta - 4T_\Delta \leq t_{so2} - 4T_\Delta \leq t_{so1} + 8T_\Delta - 4T_\Delta$$

$$t_{so1} + 2T_\Delta \leq t_{so2\_reversed\_inputs} \leq t_{so1} + 4T_\Delta \qquad (4.10)$$

A summary of the information contained in Equations (4.8) and (4.10) would be useful in order to make a conclusion regarding the accuracy of Equation (4.9). Figure 4.10 serves as such a summary.

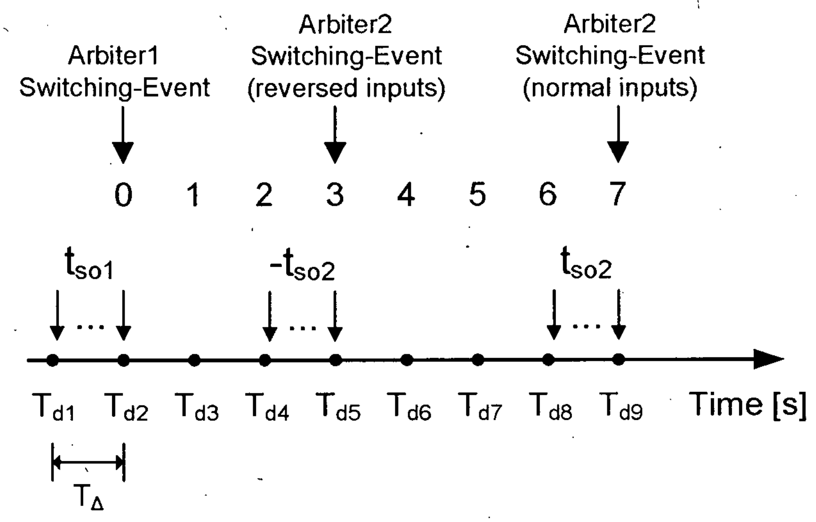


Figure 4.10: Summary of the information obtained by counting the number of oscillator cycles elapsed between the switching-events of two arbiters.

Through inspection of Figure 4.10 it can be seen that different values of t_{so2} may satisfy the relationships that have been found to exist between *Arbiter1* and *Arbiter2*. In fact, a range of t_{so2} values exist which satisfy the aforementioned relationships. In order to quantify the accuracy of Equation (4.9), the worst-case error in the estimation of t_{so2} must be determined. The worst-case error occurs when the range of t_{so2} values which satisfies the aforementioned relationships is maximized. The maximum and minimum difference between t_{so2} and $-t_{so2}$ can be found through inspection of Figure 4.10, as shown below.

$$\begin{gathered} 3T_{\Delta} < t_{so2} - t_{so2\_reversed\_inputs} < 5T_{\Delta} \\ 3T_{\Delta} < t_{so2} - (-t_{so2}) < 5T_{\Delta} \\ 1.5T_{\Delta} < t_{so2} < 2.5T_{\Delta} \end{gathered} \tag{4.11}$$

Comparison with the known value of $2T_{\Delta}$ reveals that the error of the simplified proposed calibration technique can be bound as shown in Equation (4.12).

$$-T_{\Delta}/2 < t_{ce} < T_{\Delta}/2 \tag{4.12}$$

These error bounds are equivalent to those of the direct calibration technique as discussed in Chapter 3, assuming a noise-free arbiter. If the RMS error of the simplified proposed calibration technique is calculated assuming uniformly distributed arbiter sampling offsets, then Equation (4.13) is produced.

$$RMS(t_{ce}) = \frac{T_\Delta}{2\sqrt{3}} \tag{4.13}$$

This result is equivalent to the RMS calibration error of the direct calibration technique, assuming noise-free arbiters as discussed in Chapter 3. Therefore, it can be concluded from the preceding analysis that for noise-free arbiters, the error of the simplified proposed calibration technique is equivalent to that of the direct calibration technique.

4.2 Non-Ideal Arbiters and Added Noise

As discussed in Chapter 3, the sampling offset of a non-ideal arbiter is not a fixed number, but is instead characterized by a Gaussian probability density function. As a result, two identical time intervals can induce a different response from the same arbiter. However, this fact can be exploited, as is done by the added noise-based calibration technique of Chapter 3, to improve the accuracy of the proposed calibration technique. For example, if Gaussian temporal noise with a standard deviation much larger than σ_t is added to the sequence of time intervals, many oscillator cycle counts can be recorded and then averaged in order to obtain a more accurate estimation of t_{so}. In such a case, Equation (4.9) must be rewritten as shown in Equation (4.14), where N refers to the number of oscillator cycle counts recorded when the arbiter's inputs are connected normally (state S = 0) or reversed (state S = 1).

$$t_{so} = \left(totalCycleCount_{S=0} - totalCycleCount_{S=1}\right)\frac{T_\Delta}{2N} \tag{4.14}$$

In order to explain this assertion it is necessary to examine Figure 4.11, which illustrates the PDF of each T_d within a sequence of T_ds. In addition, the sampling offsets of two arbiters, t_{so1} and t_{so2}, have been plotted along the x-axis. Note that both Figure 4.11 and the analysis to follow are predicated on the assumption that the Gaussian temporal noise added to each time interval is much greater than the intrinsic temporal noise within each arbiter, as explained in section 3.4 and depicted in Figure 3.22. Consequently, the sampling offset of an arbiter can be treated as a constant value.

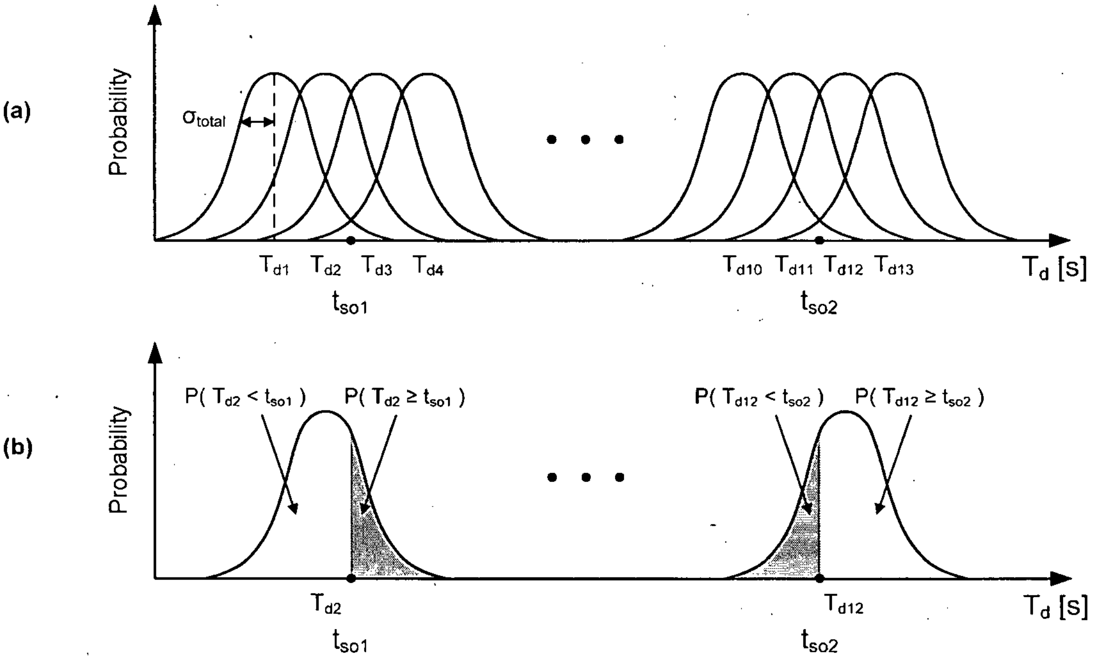


Figure 4.11: (a) PDF of several T_ds belonging to a sequence of T_ds (b) PDF of two T_ds (Note: the sampling offsets of two arbiters, t_{so1} and t_{so2}, are plotted along the x-axis of both figures).

Through inspection of Figure 4.11 (a) it can be seen that the sampling offsets of *Arbiter1* and *Arbiter2* are constant, while the instantaneous value of each time interval, i.e., T_{d1} to T_{d13}, is nondeterministic. As a result, it is not possible to predict with absolute certainty the response of either arbiter to any given time interval. Therefore, the number of oscillator cycles elapsed between the switching-events of these two arbiters is also nondeterministic. This phenomenon is the direct result of the addition of Gaussian temporal noise to each time interval.

However, if a sufficient number of oscillator cycle counts are recorded, then the statistical properties of the added temporal noise can be leveraged to obtain a very accurate estimation of t_{so}. For example, if a histogram of each T_d is compiled while the oscillator cycle counts are recorded, the histograms will more closely resemble a Gaussian distribution as time progresses. If the histograms can be accurately modelled with a Gaussian PDF, then it is possible to mathematically describe the probability of occurrence for any oscillator cycle count. To further explain, Figure 4.11 (b) may be of use. This figure depicts the probability with which T_{d2} is greater than or equal to t_{so1}, as well as the probability with which T_{d12} is greater than or equal to t_{so2}. With this sort of information it is possible to construct an equation to determine the probability of any oscillator cycle count value.

In order to construct such an equation it is first necessary to determine the probability of the oscillator cycle counter being triggered by each T_d. In the case of T_{d1} this amounts to the probability that T_{d1} is greater than or equal to t_{so1}, or $\mathrm{P}(T_{d1} \geq t_{so1})$. The equation which describes the probability of the oscillator cycle counter being triggered by T_{d2} is only slightly more complicated. The probability of such an event is equal to the probability that T_{d2} is greater than or equal to t_{so1} and T_{d1} is less than t_{so1}. Therefore, this equation is dependent upon the outcome of two distinct events, i.e., T_{d2} being greater than or equal to t_{so1} and T_{d1} being less than t_{so1}. However, since these events are independent, i.e., the value of T_{d2} is not dependent upon whether or not T_{d1} was less than t_{so1} since the added temporal noise is random, their probabilities can be dealt with individually. Mathematically these events are known as statistically independent variables [35, 36], and the probability of these two events occurring can be solved as shown in Equation (4.15), where A and B represent two independent events, and $\mathrm{A} \cap \mathrm{B}$ is the mathematical intersection of these two events.

$$\mathrm{P}(\mathrm{A} \cap \mathrm{B}) = \mathrm{P}(\mathrm{A}) \cdot \mathrm{P}(\mathrm{B}) \tag{4.15}$$

Therefore to summarize, the probability of the oscillator cycle counter being triggered by T_{d1} or T_{d2} is formally expressed in Equations (4.16) and (4.17), respectively.

$$\text{P}(T_{d1}\text{ triggers}) = \text{P}(T_{d1} \geq t_{so1}) \tag{4.16}$$

$$\begin{aligned}\text{P}(T_{d2}\text{ triggers}) &= \text{P}(T_{d2} \geq t_{so1} \cap T_{d1} < t_{so1}) \\ &= \text{P}(T_{d2} \geq t_{so1}) \cdot \text{P}(T_{d1} < t_{so1})\end{aligned} \tag{4.17}$$

With this information in hand it is now possible to construct Equation (4.18), which can be used to determine the probability of the oscillator cycle counter being triggered by any T_d, which may be represented as $T_{d(j)}$. This equation follows directly from the preceding analysis.

$$P(T_{d(j)}\ triggers) = P(T_{d(j)} \geq t_{so1}) \cdot \prod_{i=1}^{j-1} P(T_{d(i)} < t_{so1}) \tag{4.18}$$

The following definitions must also accompany Equation (4.18).

$$P(T_d \geq t_{so}) = \frac{1}{2}\left[1 + erf\left(\frac{T_d - t_{so}}{\sigma_{total}\sqrt{2}}\right)\right]$$

$$\text{P}(T_d < t_{so}) = 1 - \text{P}(T_d \geq t_{so})$$

Now that it is possible to determine the probability of the oscillator cycle counter being triggered by any T_d, it is also possible to construct an equation to determine the probability of a specific oscillator cycle count. Using an oscillator cycle count of 1 as an example, many different scenarios can be constructed to achieve this value. For example, the counter could be triggered by T_{d1} and stopped by T_{d2}, thus producing a count of 1. It is also possible for the counter to be triggered by T_{d2} and stopped by T_{d3}, therefore producing the same result. In fact, it is theoretically possible for any T_d and its successor to start and stop the counter respectively. This fact is described in Equation (4.19), where the mathematical union of these individual events is represented with the symbol 'U'.

$$\text{P(Count = 1)} = \text{P(} [T_{d1} \text{ triggers} \cap T_{d2} \text{ stops}] \cup [T_{d2} \text{ triggers} \cap T_{d3} \text{ stops}] \ldots \cup [T_{dM-1} \text{ triggers} \cap T_{dM} \text{ stops}] \text{)} \quad (4.19)$$

While Equation (4.19) is mathematically concise, it is of no use in predicting the probability of a specific oscillator cycle count unless the union of the events described in this equation are known. In order to solve the aforementioned equation, the mathematical definition of the union of two events must first be understood. Shown in Equation (4.20) is the definition of the mathematical union operation.

$$\text{P(AUB)} = \text{P(A)} + \text{P(B)} - \text{P(A}\cap\text{B)} \quad (4.20)$$

However, Equation (4.20) can be simplified by noting that each of the events described in Equation (4.19) are mutually exclusive, i.e., it is only possible for one event to occur for any single oscillator cycle count, and thus the intersections of these events are required to be zero. For example, if event A represents the scenario that T_{d1} triggers the oscillator cycle counter while T_{d2} stops it, it is not possible for some other event B to occur, which may represent the scenario under which T_{d2} triggers the oscillator cycle counter and T_{d3} stops it. This can be reasoned logically by observing that, for example, if T_{d1} is to trigger the oscillator cycle counter then it is not possible for T_{d2} or any other T_d to re-trigger the counter until it has been stopped. Similarly, if T_{d2} stops the oscillator cycle counter then it is not possible for T_{d3} or any other T_d to stop the counter until it has been re-triggered. Therefore, events A and B can be said to be mutually exclusive since they can never occur simultaneously, and Equation (4.20) can be simplified to Equation (4.21), which describes the probability of the union of two mutually exclusive events.

$$\text{P(AUB)} = \text{P(A)} + \text{P(B)} \quad (4.21)$$

With the preceding information in hand it is now possible to simplify Equation (4.19), which describes the probability of an oscillator cycle count of 1. Equation (4.22) is the result of this simplification.

$$\begin{aligned} \text{P(Count = 1)} = \; & \text{P([}T_{d1}\text{ triggers} \cap T_{d2}\text{ stops]} \cup \text{[}T_{d2}\text{ triggers} \cap T_{d3}\text{ stops]} \ldots \\ & \cup \text{[}T_{dM-1}\text{ triggers} \cap T_{dM}\text{ stops])} \\ = \; & \text{P(}T_{d1}\text{ triggers)}\cdot\text{P(}T_{d2}\text{ stops) + P(}T_{d2}\text{ triggers)}\cdot\text{P(}T_{d3}\text{ stops)} \ldots + \\ & \text{P(}T_{dM-1}\text{ triggers)}\cdot\text{P(}T_{dM}\text{ stops)} \end{aligned} \qquad (4.22)$$

Finally, as shown in Equation (4.23), it is possible to write a simplified general equation to determine the probability of any oscillator cycle count. The symbol 'M' in Equation (4.23) represents the index of the last T_d that is applied to the array of arbiters.

$$P(Count = j) = \sum_{i=1}^{M-j} P(T_{d(i)}\ triggers) \cdot P(T_{d(i+j)}\ stops) \qquad (4.23)$$

The equation which describes the probability of a specific T_d stopping the oscillator cycle counter is given in Equation (4.24). This follows directly from the analysis used while writing the equation that specifies the probability of triggering the oscillator cycle counter, which for the sake of convenience has been reproduced below.

$$P(T_{d(i+j)}\ stops) = P(T_{d(i+j)} \geq t_{so2}) \cdot \prod_{k=i+1}^{i+j-1} P(T_{d(k)} < t_{so2}) \qquad (4.24)$$

$$P(T_{d(i)}\ triggers) = P(T_{d(i)} \geq t_{so1}) \cdot \prod_{k=1}^{i-1} P(T_{d(k)} < t_{so1}) \qquad (4.25)$$

Now that it is possible to determine the probability of any oscillator cycle count, this information can be used to plot the probability of a range of oscillator cycle counts when the inputs to *Arbiter2* are both reversed and normal, as shown in Figure 4.12.

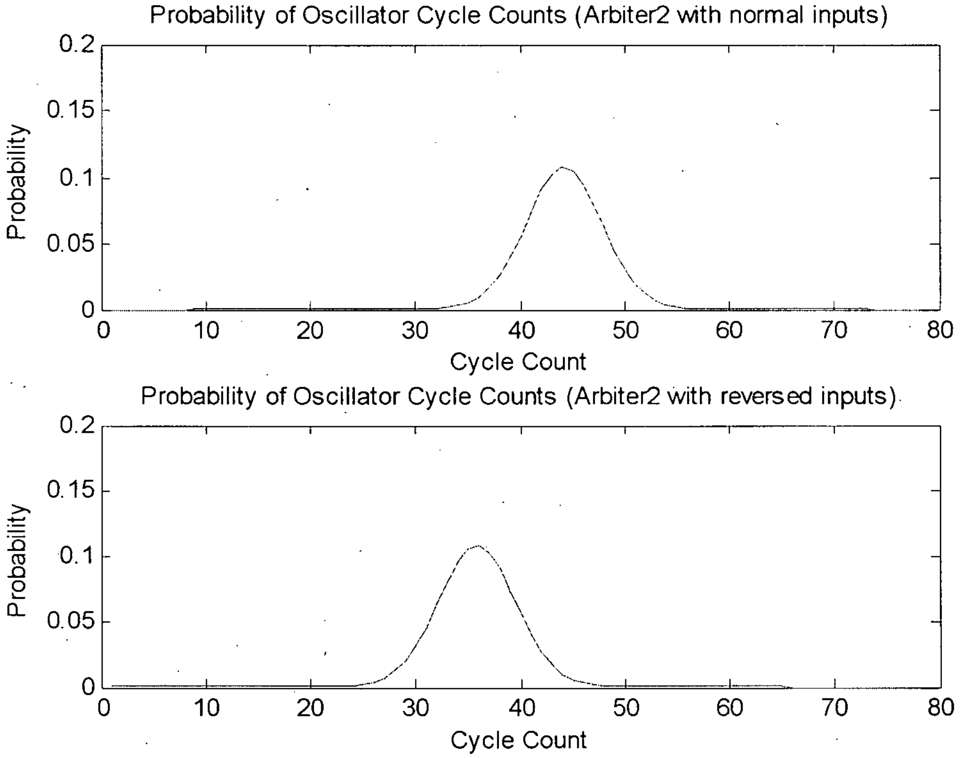


Figure 4.12: Probability of oscillator cycle counts when Arbiter2 has both normal and reversed inputs.

Through inspection of Figure 4.12 it can be seen that the PDFs of the oscillator cycle counts closely resemble Gaussian PDFs, which as stated earlier is the direct result of the addition of Gaussian temporal noise to the applied time intervals. Therefore, a good approximation of the oscillator cycle count when the inputs to *Arbiter2* are either reversed or normal can be obtained by calculating the mean of the appropriate PDF. If these mean oscillator cycle counts are known, then estimating twice the sampling offset of *Arbiter2* is a matter of calculating the difference in these mean values and multiplying by T_Δ. The result is written in Equation (4.26), where state $S = 0$ indicates that the arbiter's inputs are connected normally, and state $S = 1$ indicates that its inputs are reversed.

$$2t_{so} = \left(\mu_{S=0} - \mu_{S=1}\right) T_\Delta \tag{4.26}$$

Equation (4.26) has been written with the assumption that the PDFs of the applied time intervals are perfect Gaussians. However, this assumption is only valid when an infinite number of time intervals are applied to the arbiters. In normal circumstances only a finite number of oscillator cycle counts can be recorded due to calibration time constraints as well as physical limitations such as the depth of the oscillator cycle counters. Therefore, to be of any practical use, Equation (4.26) must be rewritten to account for these realities. Such an equation has been written in Equation (4.27), where N refers to the number of oscillator cycle counts recorded when $S = 0$ or 1.

$$2t_{so} = \left[\left(\frac{totalCycleCount_{S=0}}{N} \right) - \left(\frac{totalCycleCount_{S=1}}{N} \right) \right] T_{\Delta} \tag{4.27}$$

Next, Equation (4.27) can be rewritten to solve for the sampling offset of an arbiter, as shown in Equation (4.28).

$$t_{so} = \left(totalCycleCount_{S=0} - totalCycleCount_{S=1} \right) \frac{T_{\Delta}}{2N} \tag{4.28}$$

This equation is identical to Equation (4.14), and is therefore the core equation of the proposed SOTDC calibration technique. An analysis of the accuracy of Equation (4.28) and the proposed SOTDC calibration technique in general will be discussed in Chapter 5.

4.3 Oscillator Non-Idealities

Until this point it has been assumed that two perfectly stable oscillators with a known frequency difference are available for use during SOTDC calibration, where T_A/T_Δ is an integer. These oscillators are necessary to produce a periodic sequence of time intervals. However, in reality it is quite difficult if not impossible to obtain a perfectly stable oscillator with a precisely known frequency. Even if it was possible to build two such oscillators, the frequency of both would need to be chosen very carefully so as to ensure that a periodic sequence of time intervals could be generated. For example, if the oscillators are named *oscA* and *oscB* with periods of T_A and T_B, respectively, then T_A/T_Δ, where $T_\Delta = T_B - T_A$, must be an integer in order to produce a periodic sequence of time intervals.

One technique that could be used to eliminate the need for the aforementioned oscillators is to lock both to the same reference frequency. For example, a single reference frequency could be used to generate both *oscA* and *oscB* by means of two Phase-Locked Loops (PLLs) with different divisors. Figure 4.13 illustrates this technique.

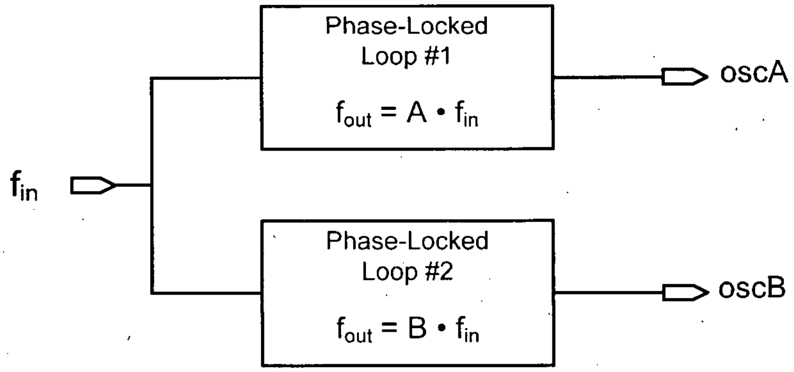


Figure 4.13: PLL implementation of *oscA* and *oscB*.

The output of the two PLLs illustrated above can be described by Equations (4.29) and (4.30).

$$f_A = A \cdot f_{in}; \quad T_A = \frac{T_{in}}{A} \tag{4.29}$$

$$f_B = B \cdot f_{in}; \quad T_B = \frac{T_{in}}{B} \tag{4.30}$$

Now, as long as Equation (4.31) yields an integer result, i.e., Q, then the sequence of time intervals produced from oscA and oscB is guaranteed to repeat itself every Q+1 time intervals.

$$\frac{T_A}{T_\Delta} = Q \tag{4.31}$$

$$T_{\Delta} = T_B - T_A \tag{4.32}$$

Substituting Equation (4.32) into Equation (4.31) produces Equation (4.33).

$$\frac{T_A}{T_B - T_A} = Q \tag{4.33}$$

Further manipulation of Equation (4.33) by means of substituting Equations (4.29) and (4.30) and then simplifying the result yields Equation (4.34).

$$Q = \frac{1}{\frac{A}{B} - 1} \tag{4.34}$$

Analysis of Equation (4.34) indicates that it is possible to generate a periodic sequence of time intervals by locking two PLLs to a single known reference frequency. The only caveat to using this technique is that A and B (the PLL divisors) must be chosen according to Equation (4.34) such that the resulting Q (the number of time intervals before the sequence repeats itself) is an integer. Further observation of this technique reveals that T_{Δ} does not need to be calibrated or measured as it can be calculated as long as f_{in} is known. Lastly, it is possible to reverse the inputs to the array of arbiters within the SOTDC by simply swapping the divisors of the two PLLs.

4.4 Implementation

The proposed SOTDC calibration technique can be implemented in a variety of ways, oftentimes requiring only a few, relatively simple, modifications to a basic SOTDC. One possible arrangement is illustrated in Figure 4.14.

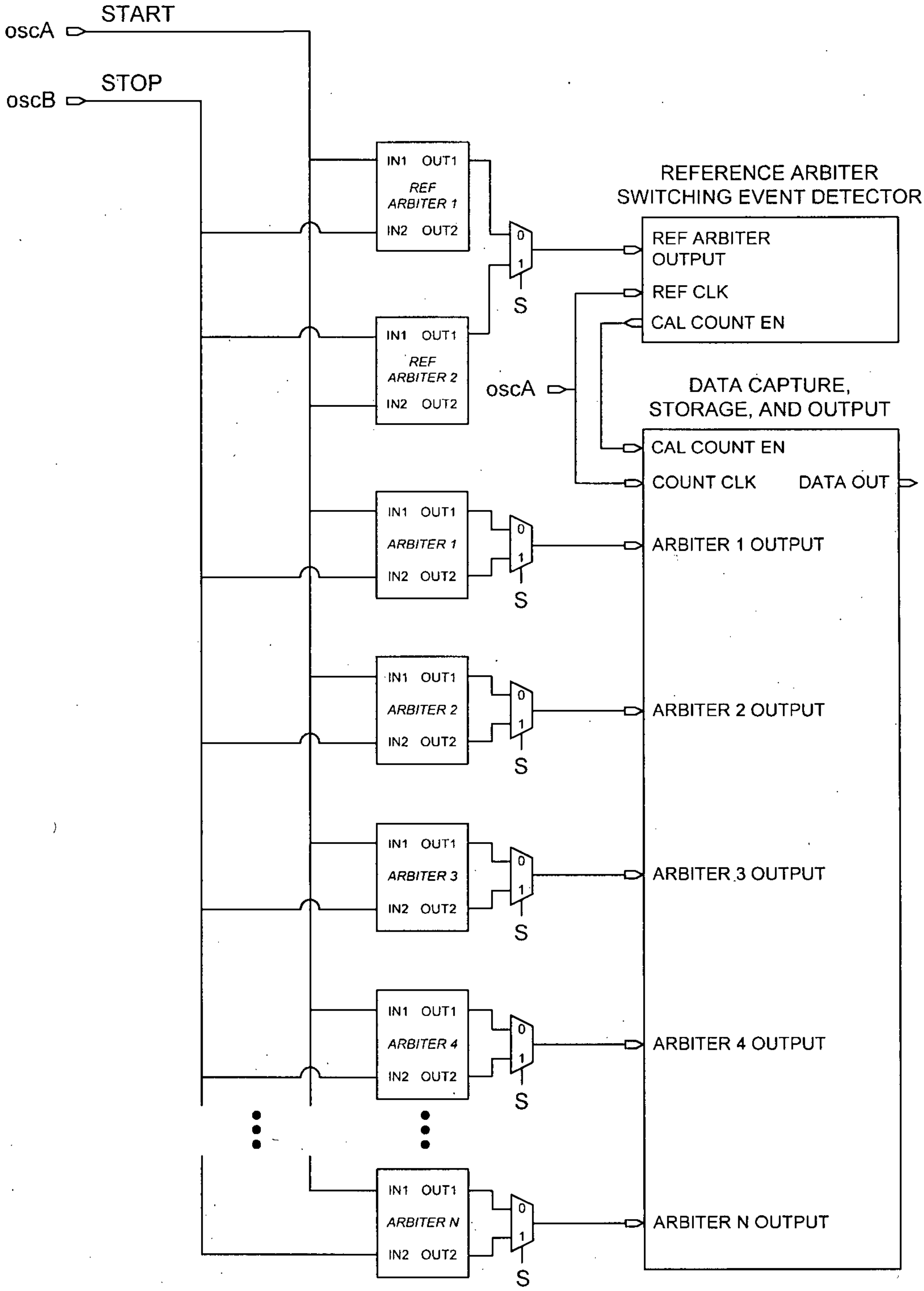


Figure 4.14: Conceptual circuit view of the proposed calibration technique.

The arbiters labelled "*ARBITER 1*" through to "*ARBITER N*" in Figure 4.14 are responsible for time interval measurement, and their sampling offsets must be calibrated. Two additional arbiters, "*REF ARBITER 1*" and "*REF ARBITER 2*", are required to realize the function of *Arbiter1*, i.e., to provide a fixed temporal reference point during calibration. As was the case for *Arbiter1*, both arbiters are required to have negative sampling offsets, and the magnitude of their sampling offsets must exceed that of any arbiter that is used for time interval measurement. Ideally, the sampling offsets of the two arbiters should be identical. In order to achieve such stringent matching requirements, the sizes of the transistors which comprise the reference arbiters must be made very large, usually ten times that of a normal arbiter. This will help to mitigate sampling offset deviations caused by process, voltage, and temperature variations.

As illustrated in Figure 4.14, the inputs to *REF ARBITER 1* are connected in an opposite manner to those of *REF ARBITER 2*. In addition, a multiplexer is used to select between the outputs of the two arbiters, indicating that only one reference arbiter is used at any given time. When S = '0', the sampling offset of *REF ARBITER 1* is a negative number. Therefore, its switching-event can be used as a temporal reference point from which it is possible to trigger an oscillator cycle counter. However, when S = '1', the sampling offset of *REF ARBITER 1* changes sign, thereby negating its role as a fixed temporal reference point. Nevertheless, it is possible to create a fixed temporal reference point if *REF ARBITER 1* and *REF ARBITER 2* are used in tandem, as the sampling offset of *REF ARBITER 2* should be nearly identical to that of *REF ARBITER 1* when S = '0' since its inputs are reversed in comparison to those of *REF ARBITER 1*. However, in order to ensure that the temporal reference point provided by *REF ARBITER 1* is nearly identical to the one provided by *REF ARBITER 2*, special attention must be paid when routing the inputs to these arbiters. Otherwise, any mismatch in the input routings may result in a significant difference between the sampling offsets of the two arbiters.

In order to detect a reference arbiter switching-event, a "*REFERENCE ARBITER SWITCHING-EVENT DETECTOR*" block is required. This switching-event detector samples the output of the reference arbiter multiplexer on the rising edge of a delayed version of *oscA*. When a switching-event is detected, the "*DATA CAPTURE, STORAGE, AND OUTPUT*" block is notified. This block consists of N oscillator cycle counters, one for each arbiter. The notification triggers the counters to begin counting on the rising edge of *oscA*. Each counter continues to increment its count until a switching-event is detected at the output of its respective arbiter. A more detailed circuit implementation of each of the aforementioned blocks can be found in Appendix A.

It is important to note that the oscillator cycle counters required by the proposed calibration technique are almost always used in a basic SOTDC to quickly store the results of a large number of time interval measurements. Therefore, these counters would not normally increase the area of an SOTDC. However, it is possible to reduce the area consumed by these counters by sharing only one amongst the *N* arbiters and calibrating each arbiter sequentially. It should also be mentioned that care must be taken when routing the *START* and *STOP* inputs of an SOTDC. Otherwise, electromagnetic coupling between these two input lines may significantly alter the oscillator cycle counts, which would adversely affect the accuracy of the proposed calibration technique. Lastly, two additional circuits not illustrated in Figure 4.14 are required for successful implementation of the proposed calibration technique. These circuits include a Gaussian noise generation circuit [37] and a voltage-controlled delay buffer [38]. Together these circuits can be used to generate and then convert Gaussian noise from the voltage to the time domain, which in turn is used to modulate the time intervals applied to an SOTDC during calibration.

4.5 Summary

The practical benefit accrued from the use of the proposed calibration technique is the ability to perform calibration without knowledge of the values of the time intervals applied to the SOTDC during calibration. Only knowledge of the temporal difference between adjacent time intervals is required. This information can be acquired through the selection of a reference frequency (f_{in}) and PLL divisors (A and B) according to Equation (4.34). In addition, there is no need to apply a curve fitting function to the calibration results. Post-processing of the results consists of simple subtraction, multiplication, and division operations. Therefore, the post-processing requirements of the proposed calibration technique are much less demanding than those of either the direct or the added noise-based calibration techniques. Chapter 5 will present a thorough analysis of the accuracy of the proposed calibration technique.

Chapter 5

Results and Analysis

The proposed SOTDC calibration technique has been presented in Chapter 4. This presentation included a discussion of the proposed calibration technique's principle of operation as well as a conceptual circuit-based implementation. In addition, an equation was presented to estimate the sampling offset of an arbiter assuming an absence of temporal noise in both the arbiters and the time intervals. A theoretical bound was then placed on the error of the proposed calibration technique given an absence of temporal noise in both the arbiters and the time intervals, and this bound was shown to be identical to that of the direct calibration technique assuming the same conditions. Next, the discussion progressed towards the consideration of thermal noise in the arbiters and Gaussian temporal noise in the sequence of time intervals. A statistically-based mathematical equation was developed to gain further insight into the operation of the proposed calibration technique in the presence of additive Gaussian temporal noise. Eventually it was shown that the equation used to estimate the sampling offset of a noise-free arbiter could also be used to estimate the sampling offset of an arbiter exposed to Gaussian temporal noise.

It was then mentioned that the practical benefit accrued from the use of the proposed calibration technique is the ability to perform calibration without knowledge of the values of the time intervals applied to the SOTDC during calibration. Only knowledge of the temporal difference between subsequent time intervals is required, i.e., T_Δ must be known. Lastly, it was shown that T_Δ does not need to be calibrated or measured as it can be calculated as long as the reference frequency and divisors of the two PLLs which generate *oscA* and *oscB* are known. However, the error of the proposed calibration technique was not addressed in the preceding discussion, as it is the topic of this chapter.

5.1 Theoretical Error Bounds

The statistically-based mathematical equations developed in Chapter 4 are reproduced below. These equations were derived in order to gain further insight into the operation of the proposed calibration technique in the presence of Gaussian temporal noise.

$$P(Count = j) = \sum_{i=1}^{M-j} P(T_{d(i)}\ triggers) \cdot P(T_{d(i+j)}\ stops)$$

$$P(T_{d(i)}\ triggers) = P(T_{d(i)} \geq t_{so1}) \cdot \prod_{k=1}^{i-1} P(T_{d(k)} < t_{so1})$$

$$P(T_{d(i+j)}\ stops) = P(T_{d(i+j)} \geq t_{so2}) \cdot \prod_{k=i+1}^{i+j-1} P(T_{d(k)} < t_{so2})$$

The symbol 'M' represents the index of the last T_d in the sequence of T_ds that is applied to the array of arbiters. In addition, the definitions of P($T_d \geq t_{so}$) and P($T_d < t_{so}$) are reproduced below, where the assumption has been made that the time intervals applied to the arbiters have been altered to fit a Gaussian distribution with a standard deviation much larger than that of the random temporal noise which is intrinsic to the arbiters, i.e., $\sigma_{added} >> \sigma_t$, and therefore, $\sigma_{total} \approx \sigma_{added}$.

$$P(T_d \geq t_{so}) = \frac{1}{2}\left[1 + erf\left(\frac{T_d - t_{so}}{\sigma_{total}\sqrt{2}}\right)\right]$$

$$\mathrm{P}(T_d < t_{so}) = 1 - \mathrm{P}(T_d \geq t_{so})$$

These equations can be used to determine the probability of any oscillator cycle count value. As stated in Chapter 4, estimating the sampling offset of an arbiter is then reduced to finding the difference in the mean values of the oscillator cycle count PDFs when the inputs to the arbiters are normal and reversed, and then multiplying the result by $T_\Delta/2$ as shown in Equation (5.1).

$$t_{so} = \left(\mu_{S=0} - \mu_{S=1}\right)\frac{T_\Delta}{2} \tag{5.1}$$

However, one would only estimate the sampling offset of an arbiter using these statistically-based mathematical equations if they wanted to determine the theoretical error bounds of the proposed calibration technique. This is true since the sampling offset of the arbiter in question is a required parameter of these equations. In addition, the sampling offset calculated using these equations is very likely to contain the smallest possible proposed SOTDC calibration technique error given a particular value of T_Δ and σ_{total}. This assertion can be explained by realizing that the PDFs of the time intervals applied to the arbiters are assumed to be perfectly Gaussian when using the aforementioned equations. However, in reality only a finite number of oscillator cycle counts can be recorded due to calibration time constraints as well as physical limitations such as the depth of the oscillator cycle counters. As a result, the PDFs of the time intervals applied to the arbiters during SOTDC calibration are not perfectly Gaussian, and it is this deviation that introduces additional error into the actual calibration results.

In an effort to ascertain the theoretically smallest proposed SOTDC calibration technique error, a Matlab model has been constructed using the statistically-based mathematical equations. This model provides the ability to calculate the minimum error of the proposed SOTDC calibration technique for any value of T_{Δ} and σ_{total}. The model has been constructed using an array of 100 arbiters uniformly distributed across one T_{Δ}. This was done to ensure that the results were independent of any particular sampling offset value. The RMS calibration error of 100 arbiters calculated using four different T_{Δ} values and numerous values of σ_{total} is shown in Figure 5.1.

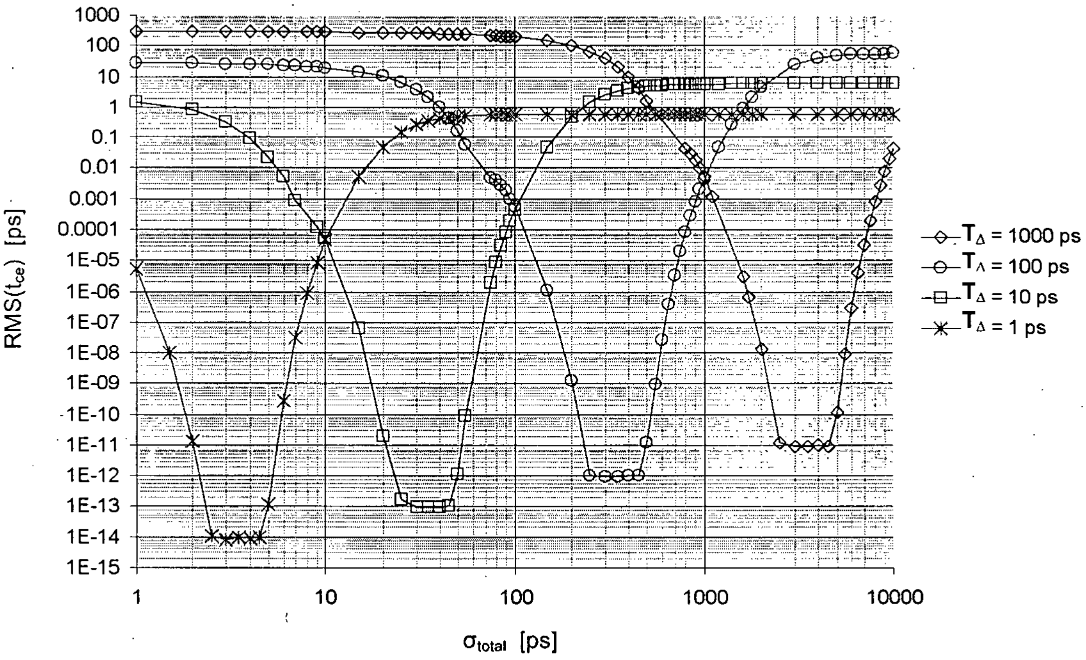


Figure 5.1: Minimum error of the proposed calibration technique across four different T_{Δ} values.

Inspection of Figure 5.1 reveals that for each value of T_{Δ} there is a range of σ_{total} values which yield a dramatically lower calibration error (t_{ce}) than do the rest. In fact, it appears as if this range of σ_{total} values yields the lowest calibration error that is achievable using the proposed calibration technique. However, this perceived calibration error floor is the result of a limitation in the numerical accuracy of the computer used to perform the calculations, and does not represent an actual limitation in the lowest achievable calibration error for a particular T_Δ value. The true limit may in fact approach zero as the standard deviation of the added temporal noise moves closer to some optimal value of σ_{total}.

Inspection of Figure 5.1 also reveals that the shape of the minimum error curve is constant across all four T_Δ values. As σ_{total} is decreased from its optimal value the minimum calibration error increases sharply. However, as σ_{total} is decreased even further, the minimum calibration error eventually saturates. While it is true that the saturation value is dependent upon the actual value of T_Δ, it can be shown that T_Δ and the saturation value scale proportionally. Analogously, as σ_{total} is increased from its optimal value the minimum calibration error increases sharply and eventually saturates to a value that is proportional to T_Δ.

In order to understand why the minimum calibration error changes the way it does as σ_{total} is varied from its optimal value, a plot of the oscillator cycle count PDFs for numerous σ_{total} values may prove useful. Such plots are shown in Figure 5.2 and Figure 5.3.

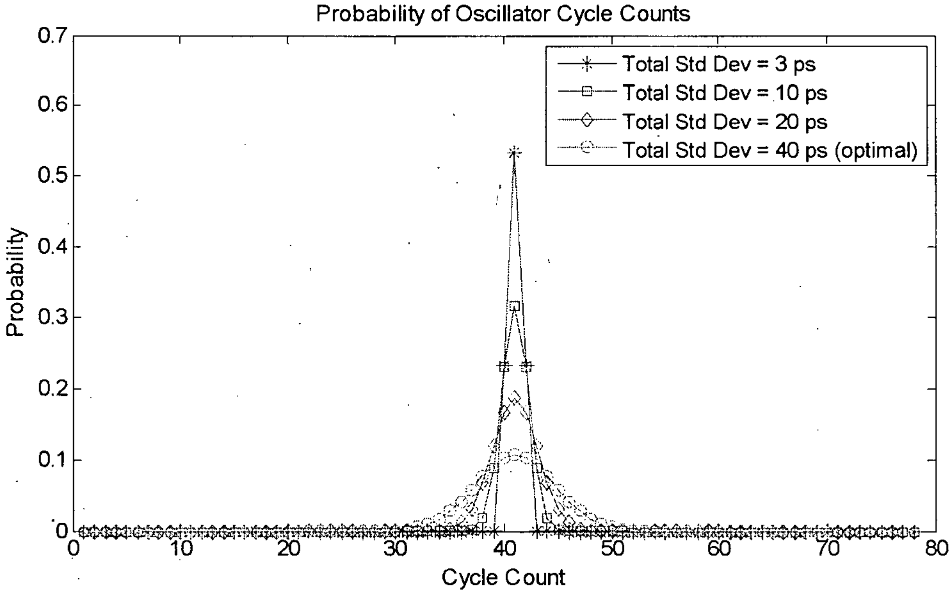


Figure 5.2: Oscillator cycle counts PDFs for several different values of σ_{total}, all of which are $\leq \sigma_{optimal}$ ($T_\Delta = 10$ ps).

Inspection of Figure 5.2 reveals that as σ_{total} is decreased from its optimal value, the oscillator cycle count PDFs become less Gaussian and begin to more closely resemble a unit impulse. In fact, if σ_{total} is reduced to 0 ps, as is the case with the simplified proposed calibration technique discussed in section 4.1, the oscillator cycle count PDF transforms into an ideal unit impulse. This makes intuitive sense as the oscillator cycle count must be a constant value when there is a complete absence of temporal noise in an arbiter. As discussed in section 4.1 and derived in section 3.2.1, the RMS calibration error of the proposed SOTDC calibration technique can be obtained using the following equation when $\sigma_{total} = 0$ ps:

$$RMS(t_{ce}) = \frac{T_{\Delta}}{2\sqrt{3}}$$

This assertion can be verified by comparing the results obtained from the preceding equation with those plotted in Figure 5.1 when $\sigma_{total} << \sigma_{optimal}$. In summary, the minimum calibration error of the proposed SOTDC calibration technique increases as σ_{total} is decreased from its optimal value. This phenomenon can be explained by observing that the mean value of an oscillator cycle count PDF approaches a whole number as σ_{total} is decreased from its optimal value, thereby increasing the calibration error for arbiters with sampling offsets that are not integer multiples of T_{Δ}.

Inspection of Figure 5.3 reveals that as σ_{total} is increased from its optimal value, the oscillator cycle count PDFs become less Gaussian and eventually converge to an exponentially decaying curve.

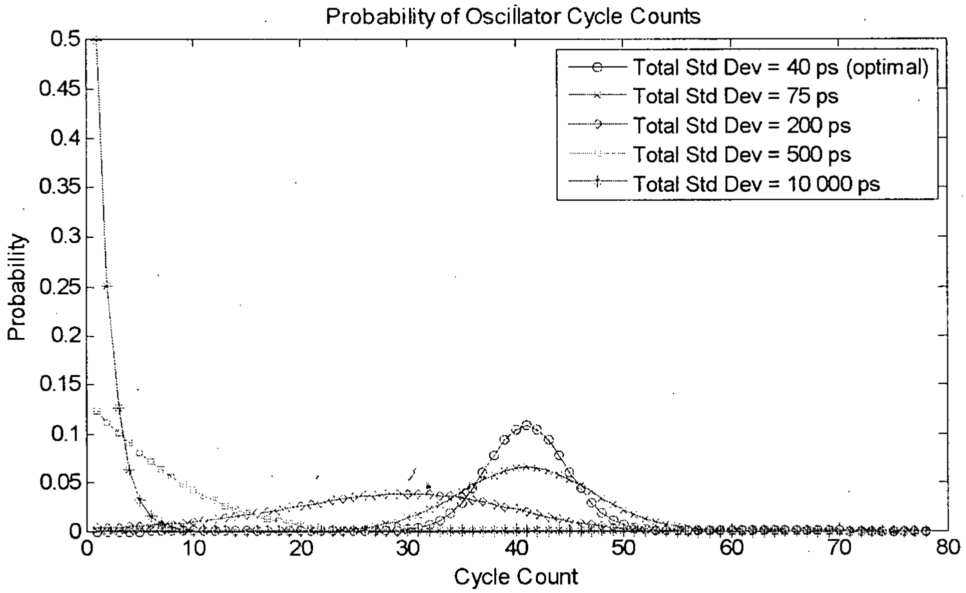


Figure 5.3: Oscillator cycle counts PDFs for several different values of σ_{total}, all of which are $\geq \sigma_{optimal}$ ($T_\Delta = 10$ ps).

Equation (5.2) has been written to mathematically describe the oscillator cycle count PDF curve when $\sigma_{total} >> |t_{so\_arbiter} - t_{so\_ref\_arbiter}|$, where $t_{so\_arbiter}$ represents the sampling offset of the arbiter under calibration and $t_{so\_ref\_arbiter}$ the sampling offset of the arbiter used as an unknown temporal reference point. The variable n represents the oscillator cycle count value.

$$P(n) = \frac{1}{2} e^{\ln(0.5)(n-1)}; \quad n > 0 \tag{5.2}$$

Equation (5.2) indicates that the probability of a particular oscillator cycle count value is exactly half that of its predecessor. It is also evident from the preceding equation that an oscillator cycle count equal to one occurs with a probability of 0.5 when $\sigma_{total} >> |t_{so\_arbiter} - t_{so\_ref\_arbiter}|$. These two facts can be explained by realizing that once the oscillator cycle counter has been triggered by a particular time interval, i.e., a time interval that is $\geq t_{so\_ref\_arbiter}$ has been applied to the arbiters, any subsequent time interval may exceed $t_{so\_arbiter}$ with a probability of 0.5. This behaviour is a direct result of the relationship between the added temporal noise and the difference between the sampling offset of the reference arbiter and that of the arbiter under calibration, i.e., $\sigma_{total} >> |t_{so\_arbiter} - t_{so\_ref\_arbiter}|$. Assuming the added temporal noise is purely random, i.e., its future behaviour is not dependent upon its past, and $\sigma_{total} >> |t_{so\_arbiter} - t_{so\_ref\_arbiter}|$, the probability of a particular oscillator cycle count can then be found using Equation (5.3).

$$\begin{aligned} P(Count = j) &= P(T_{d(j)} \geq t_{so\_arbiter}) \cdot \prod_{k=1}^{j-1} P(T_{d(k)} < t_{so\_arbiter}) \\ &= \left(\frac{1}{2}\right)^{j} ; \ j > 0 \end{aligned} \tag{5.3}$$

Returning to the discussion of the shape of the minimum calibration error curve (Figure 5.1) when $\sigma_{total} \geq \sigma_{optimal}$, it is now possible to predict the value at which the curve saturates. For example, it is now understood that when $\sigma_{total} >> |t_{so\_arbiter} - t_{so\_ref\_arbiter}|$, the shape of the oscillator cycle count PDF is always an exponentially decaying curve. In fact, this is true regardless of the orientation of the arbiter's inputs, i.e., normal or reversed, as the logic presented in the previous paragraph applies to either situation. To be more precise, as long as $\sigma_{total} >> |t_{so\_arbiter} - t_{so\_ref\_arbiter}|$ holds true, then the oscillator cycle count PDF is always an exponentially decaying curve that can be described using Equation (5.2) or (5.3). However, this infers that the estimation of the arbiter's sampling offset is always equal to zero, as shown below, where $S = 0$ indicates the arbiter's inputs are connected in a normal manner, and $S = 1$ indicates that they are reversed.

$$t_{so} = (\mu_{S=0} - \mu_{S=1})\frac{T_{\Delta}}{2} = (0)\frac{T_{\Delta}}{2} = 0$$

If the arbiter's estimated sampling offset converges to zero as σ_{total} is increased to the point where it is $>> |t_{so\_arbiter} - t_{so\_ref\_arbiter}|$, then the calibration error must saturate at the value of the arbiter's sampling offset. In fact, this can be observed in Figure 5.1, where each curve saturates at the RMS value of the sampling offsets of the 100 arbiters specified in the aforementioned Matlab model. The saturation value changes from one curve to the next since the sampling offsets of the arbiters are variable as they have been intentionally chosen to always span one T_{Δ}. This ensures that the results are independent of any particular sampling offset value. Further inspection of Figure 5.1 corroborates the assertion that the saturation value of the calibration error is equal to the RMS value of the arbiters' sampling offsets, as it can be seen that the saturation values are proportional to T_{Δ}.

At this point it is instructive to remind the reader that the recommended implementation of the proposed SOTDC calibration technique does not employ alternate routing in order to reverse an arbiter's inputs. Instead, the divisors of the two PLLs used to produce the calibration oscillators are swapped to achieve the same effect.

5.2 Realistic Error Bounds

Now that the theoretical capabilities of the proposed calibration technique have been presented, it is possible to discuss the accuracy of the proposed calibration technique under a more realistic set of conditions. Under such a scenario, the assumption that the PDFs of the time intervals are perfect Gaussians is no longer justified, as a practical SOTDC calibration technique must operate within a finite amount of time. A restriction on the total calibration time places a limit on the number of measurements performed during calibration, or in the case of the proposed calibration technique, results in a finite number of oscillator cycle counts that can be recorded. A finite number of oscillator cycle counts translates into time interval PDFs that are no longer ideal Gaussians, and as these time intervals become less Gaussian, so then do the PDFs of the oscillator cycle counts. Finally, as the PDFs of the oscillator cycle counts become less Gaussian, a greater error is introduced into the arbiter sampling offset estimation. This is true since a component of the equation used to estimate the sampling offset of an arbiter involves calculating the mean value of the oscillator cycle counts when the inputs to the arbiter are both reversed and normal, as shown in Equation (5.4).

$$t_{so} = \left[\left(\frac{totalCycleCount_{S=0}}{N} \right) - \left(\frac{totalCycleCount_{S=1}}{N} \right) \right] \frac{T_{\Delta}}{2} \qquad (5.4)$$

It is possible to test the assertion that the error of the proposed SOTDC calibration technique should decrease as the PDFs of the oscillator cycle counts become more Gaussian. For example, it has been shown that the PDF of a random variable approaches that of a Gaussian distribution as the number of trials increases [39]. Therefore, it would seem logical for the error of the proposed SOTDC calibration technique to decrease as the number of measurement repetitions is increased. In an effort to validate this theory, a behavioural Matlab model representation of the proposed SOTDC calibration technique has been constructed. The decision to construct the model using Matlab as opposed to a circuit-based simulation environment was made on the basis of simulation time. The Matlab model was found to execute simulations up to 100 times faster than the circuit-based model. As the circuit-based model did not provide any additional insight into the capabilities of the proposed SOTDC calibration technique, it was decided to collect all data using the Matlab model.

The input to the aforementioned Matlab model consists of the following four parameters:

- The number of arbiters in the SOTDC.
- T_{Δ}: the temporal difference in the periods of the two oscillators that are used to calibrate the SOTDC.
- σ_{total}: the standard deviation of an arbiter's sampling offset. This number includes the Gaussian temporal noise that is added to the output of one of the calibration oscillators, in addition to the Gaussian temporal noise that is intrinsic to each arbiter.
- N: the number of measurement repetitions.

These four parameters can be used to predict the RMS calibration error (t_{ce}) of the estimated sampling offsets of an array of arbiters. To ensure that the results of this model are independent of any particular sampling offset value, the sampling offsets of the arbiters have been uniformly distributed across one T_{Δ}. The model's method of operation can be described as illustrated in Figure 5.4.

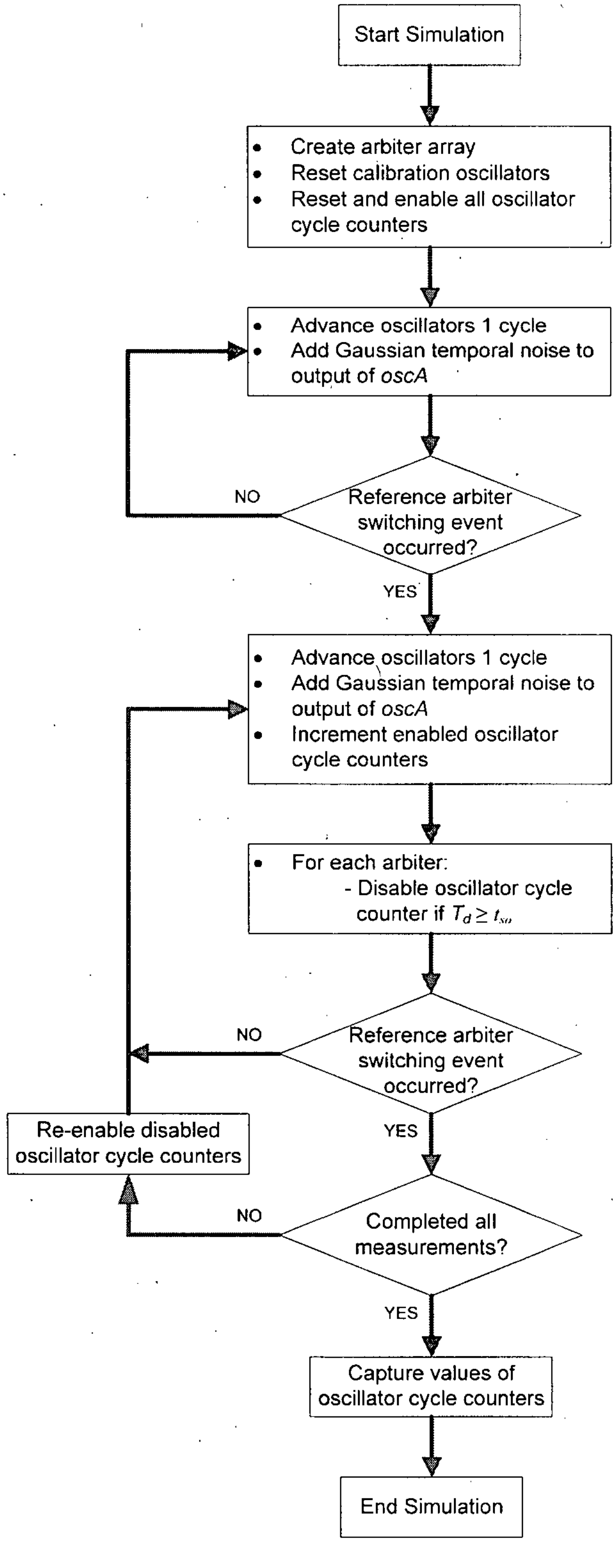

Start Simulation
Create arbiter array
Reset calibration oscillators
Reset and enable all oscillator cycle counters
Advance oscillators 1 cycle
Add Gaussian temporal noise to output of *oscA*
NO
Reference arbiter switching event occurred?
YES
Advance oscillators 1 cycle
Add Gaussian temporal noise to output of *oscA*
Increment enabled oscillator cycle counters
For each arbiter:
- Disable oscillator cycle counter if $T_d \geq t_{su}$
NO
Reference arbiter switching event occurred?
YES
Re-enable disabled oscillator cycle counters
NO
Completed all measurements?
YES
Capture values of oscillator cycle counters
End Simulation

Figure 5.4: Flowchart describing the method of operation of the proposed SOTDC calibration technique Matlab model.

Each Matlab simulation produces an array of oscillator cycle counter values, where the length of the array is equal to the number of arbiters in the simulated SOTDC. In order to obtain the RMS calibration error of the estimated sampling offsets, two distinct simulations must be performed, each consisting of N measurement repetitions. The first simulation is performed with the SOTDC's oscillator inputs connected in a normal manner, while the other with the inputs effectively reversed. Therefore, the first simulation yields the number of oscillator cycles elapsed between the switching-events of the reference arbiter and the arbiter under calibration, while the second simulation yields the number of oscillator cycles elapsed when the inputs to the arbiter under calibration are reversed. The two oscillator cycle counter value arrays produced by these simulations can then be used to calculate the estimated arbiter sampling offsets. Equation (4.28), which for the sake of convenience has been reproduced below, should be used to compute the estimated arbiter sampling offsets.

$$t_{so} = \left(totalCycleCount_{S=0} - totalCycleCount_{S=1}\right)\frac{T_{\Delta}}{2N}$$

Lastly, the RMS calibration error can be determined using the array of estimated arbiter sampling offsets.

As stated earlier, the error of the proposed SOTDC calibration technique is expected to decrease as the number of measurement repetitions is increased. This conjecture was formed based on two interrelated assumptions; the first of which states that the mean value of an oscillator cycle count PDF will deviate from that of an ideal Gaussian, even when an optimal value of σ_{total} is used. The second assumption states that this deviation should decrease as the number of measurement repetitions is increased. Fortunately, it is now possible to validate this conjecture using the Matlab model of the proposed SOTDC calibration technique.

The aforementioned model has been used to produce three oscillator cycle count histograms, each of which have been generated using a different number of measurement repetitions. In addition, an optimal value of σ_{total} has been used, while only one arbiter was simulated. These histograms are shown in Figure 5.5.

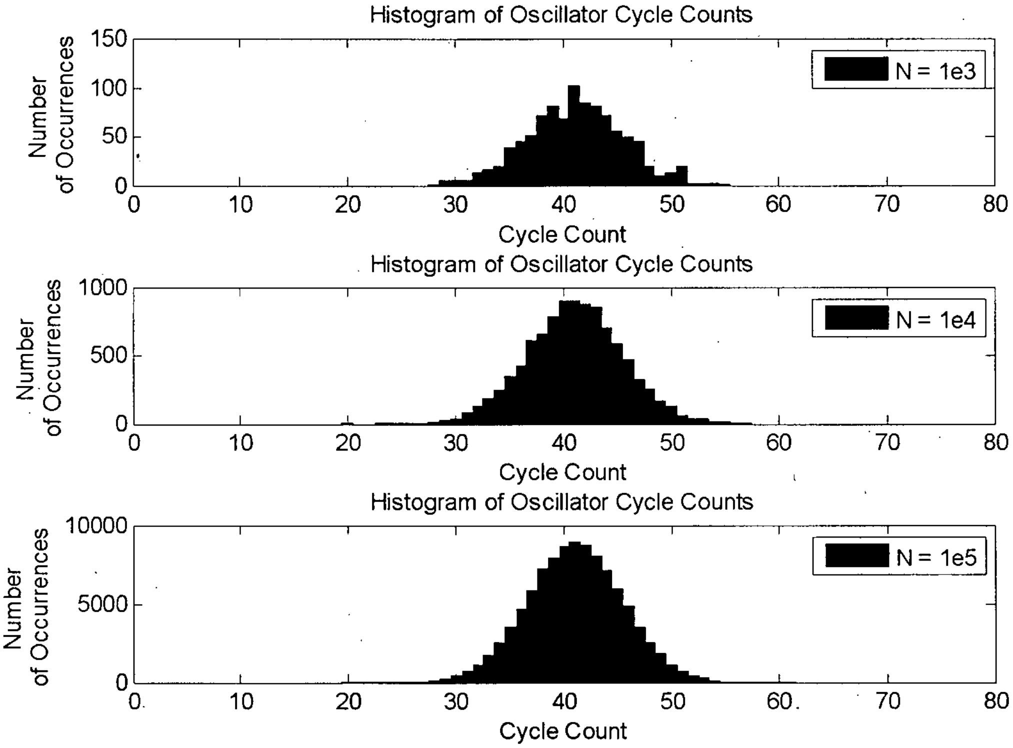


Figure 5.5: Three oscillator cycle count histograms, generated using a different number of measurement repetitions.

As it is difficult to visibly discern the mean values of these histograms, their pertinent properties have been compiled into Table 5.1.

Table 5.1: Properties of the oscillator cycle count histograms illustrated in Figure 5.5.

N	Mean [ps]	Standard Deviation [ps]	Error [ps]
1 000	41.2560	4.440	0.5150
10 000	41.0178	4.450	0.3135
100 000	41.0037	4.547	0.0184

Analysis of the results presented in Table 5.1 reveals that while increasing the number of measurement repetitions from 1 000 to 100 000 doesn't appear to drastically alter the mean value or standard deviation of the oscillator cycle count histograms, the RMS error of the estimated arbiter sampling offset is observed to decrease by more than a factor of twenty-five. In addition, the histograms can be seen to converge towards a Gaussian distribution as the number of measurement repetitions is increased. In summary, the results of Table 5.1 demonstrate that relatively small deviations in the mean value of an oscillator cycle count PDF can have a significant impact on the accuracy of the estimated value of an arbiter sampling offset. However, the magnitude of these deviations, and hence the error they introduce into the estimated arbiter sampling offset, can be drastically decreased by increasing the number of measurement repetitions.

Now that the Matlab model of the proposed SOTDC calibration technique has been shown to yield sensible results, it is possible to perform a more thorough error analysis. For example, it is now possible to compare the error of the proposed calibration technique, given a certain number of measurement repetitions, with the minimum error calculated using the statistically-based mathematical equations presented in section 5.1. One way to perform this comparison would involve plotting the realistic error of the proposed calibration technique alongside the theoretical results of Figure 5.1. Indeed, such a plot has been created, as shown in Figure 5.6, where the RMS calibration error of 100 arbiters have been calculated using numerous values of σ_{total} while T_Δ has been fixed at 10 ps.

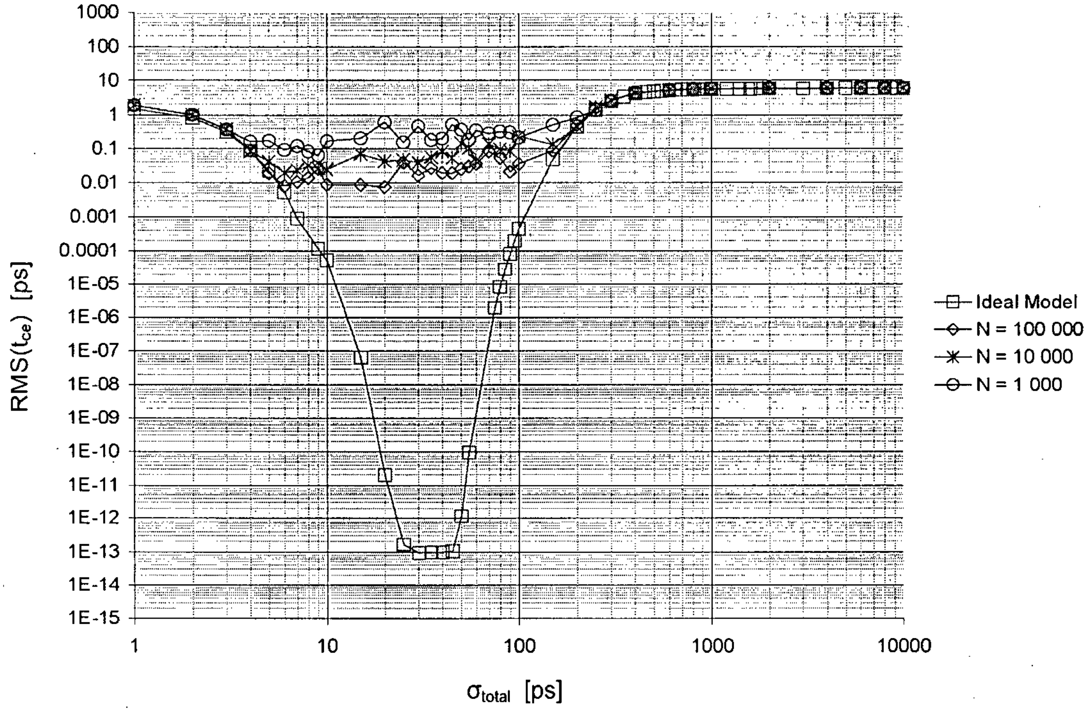


Figure 5.6: Comparison of the theoretical and realistic RMS error of the proposed calibration technique when $T_{\Delta} = 10$ ps.

Through inspection of Figure 5.6 it is possible to form numerous insights regarding the capabilities of the proposed SOTDC calibration technique in addition to the correctness of the two Matlab models that have been created in order to predict its behaviour. Firstly, it can be observed that the realistic and theoretical calibration errors saturate at the same value when σ_{total} is decreased or increased from its optimal value. This fact serves to further increase the credibility of the two Matlab models, which together predict the theoretical and practical capabilities of the proposed SOTDC calibration technique, as these models employ completely different algorithms. Secondly, it can be observed that both models produce similar optimal values of σ_{total}. However, the range of optimal σ_{total} values produced by the realistic model is wider than that of the theoretical model. This can be explained by realizing that a practical implementation of the proposed SOTDC calibration technique is subject to two different sources of error, whereas a theoretical implementation is only subject to one.

For example, both models predict that the calibration error will increase as σ_{total} is moved farther from its optimal value. The effects of a non-optimal σ_{total} value on an oscillator cycle count PDF are depicted in Figure 5.2 and Figure 5.3. However, only the realistic model is capable of predicting the effects of a finite number of measurement repetitions on an oscillator cycle count PDF, and hence the calibration results, as depicted in Figure 5.5. Therefore, as σ_{total} is brought close to its optimal value, the realistic model predicts that the largest contributor to an arbiter's calibration error is the finite number of measurement repetitions. As a result, arbiter calibration performed using a finite number of measurement repetitions is shown to produce a much higher error than is otherwise theoretically possible. Conversely, as σ_{total} is moved farther from is its optimal value, an arbiter's calibration error is predicted to be dominated by the non-optimal σ_{total} value.

Perhaps the most useful insight that can be learned through inspection of Figure 5.6 concerns the relationship between the RMS error of the estimated arbiter sampling offsets and the number of measurement repetitions. It can be seen that in general, increasing the number of measurement repetitions does indeed decrease the RMS error of the estimated arbiter sampling offsets. However, since the temporal noise that is added to the output of one of the calibration oscillators is random in nature, and therefore cannot be guaranteed to conform to a Gaussian distribution over finite time intervals, it is possible for the RMS error to actually increase as the number of measurement repetitions is increased. Nevertheless, the probability of this scenario quickly diminishes as the number of measurement repetitions is further increased. This assertion can be supported by examining Figure 5.7, which depicts the three non-ideal curves shown in Figure 5.6 over an optimal range of σ_{total} values. Also shown in Figure 5.7 are three linear approximations to the aforementioned data sets.

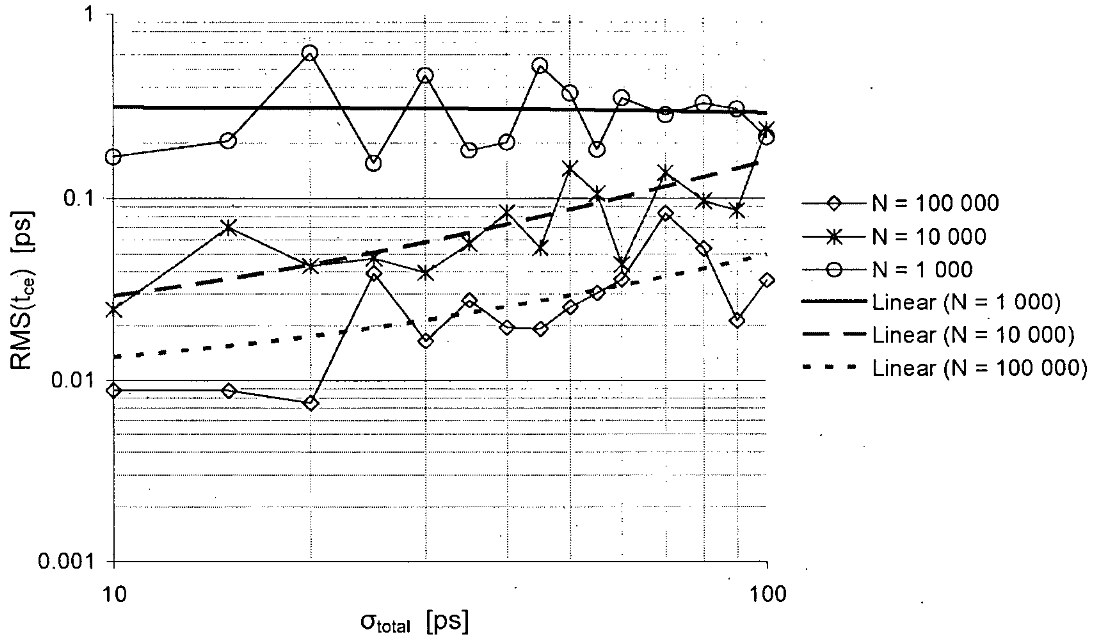


Figure 5.7: The RMS error of the proposed calibration technique when $T_\Delta = 10$ ps, plotted for three different number of measurement repetitions (N).

While it can be observed from Figure 5.7 that it is possible for a tenfold increase in the number of measurement repetitions to have little positive effect on the RMS calibration error of the arbiter sampling offsets, it can also be seen that a one hundredfold increase will almost certainly provide a substantial reduction in this error. For example, if the linear approximations are evaluated at $\sigma_{total} = 30$ ps, which is the optimal value of σ_{total} predicted by the statistically-based model, the resulting RMS calibration errors are approximately 0.3 ps, 0.06 ps, and 0.02 ps when N = 1 000, 10 000, and 100 000, respectively. Therefore, a fifteen fold reduction in the RMS calibration error of the arbiter sampling offsets is observed when the number of measurement repetitions is increased by a factor of one hundred. However, it should be reiterated that due to the random nature of the temporal noise that is used to vary the oscillator time intervals, this improvement may vary significantly from one calibration to the next. While it is possible to reduce this random variation by fitting the oscillator cycle counts histogram to a Gaussian PDF curve before calculating its mean, similar to what is performed as part of the Added Noise calibration technique of section 3.4, this would require a far more onerous post-processing step without contributing a meaningful improvement in calibration accuracy. As a final observation of Figure 5.7, it can be seen that it is possible to achieve an RMS calibration error well below 0.1 ps when T_{Δ} = 10 ps through increasing the number of measurement repetitions to 100 000.

In order to understand the repeatability of the aforementioned results and their reliance on the chosen value of T_{Δ}, three additional plots have been created, each using a different value of T_{Δ}. Figure 5.8 depicts four data sets which are very similar to the ones shown in Figure 5.7, however in this case T_{Δ} has been increased to 100 ps. Three key observations can be made from inspection of Figure 5.8. Firstly, increasing the number of measurement repetitions from 1 000 to 100 000 is observed to decrease the RMS calibration error by approximately a factor of ten. This result is in agreement with what was observed when T_{Δ} = 10 ps. Secondly, a tenfold increase in the RMS calibration error is observed across all three measurement repetition values when compared with the results obtained when T_{Δ} = 10 ps. Thirdly, and following directly from the two previous observations, approximately the same RMS calibration error is achieved with 100 000 measurement repetitions when T_{Δ} = 100 ps as is achieved with 1 000 measurements when T_{Δ} = 10 ps. Therefore, as T_{Δ} is increased by a factor of ten, the number of measurement repetitions must be increased by a factor of one hundred in order to maintain the same calibration accuracy.

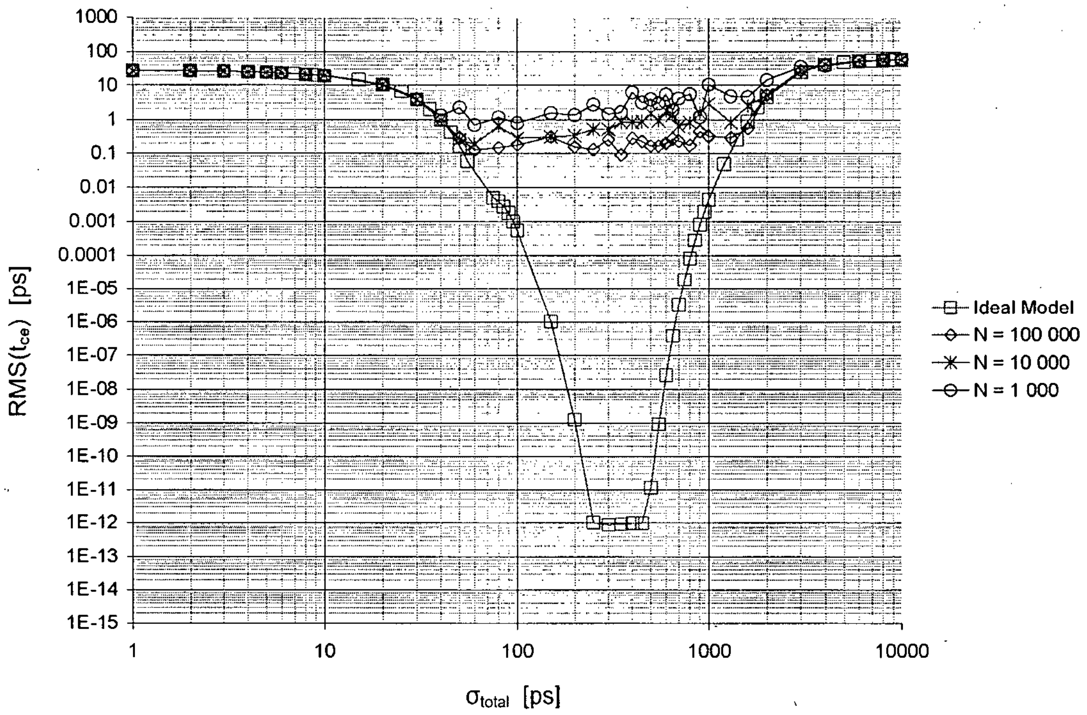


Figure 5.8: Comparison of the theoretical and realistic RMS error of the proposed calibration technique when T_{Δ} = 100 ps.

Depicted in Figure 5.9 are four additional data sets produced when T_{Δ} is further increased to 1 ns. Once again, the same three key observations can be made. For example, increasing the number of measurement repetitions by a factor of one hundred is shown to yield nearly a ten fold decrease in RMS calibration error. However, the predicted RMS calibration error is still approximately ten times greater than what was shown to be possible when T_{Δ} = 100 ps, assuming the same number of measurement repetitions are performed. Lastly, in order to achieve the same RMS calibration error as was predicted when T_{Δ} = 100 ps, the number of measurement repetitions must be increased by a factor of one hundred. Figure 5.10 depicts the RMS calibration error of the proposed technique when T_{Δ} = 1 ps.

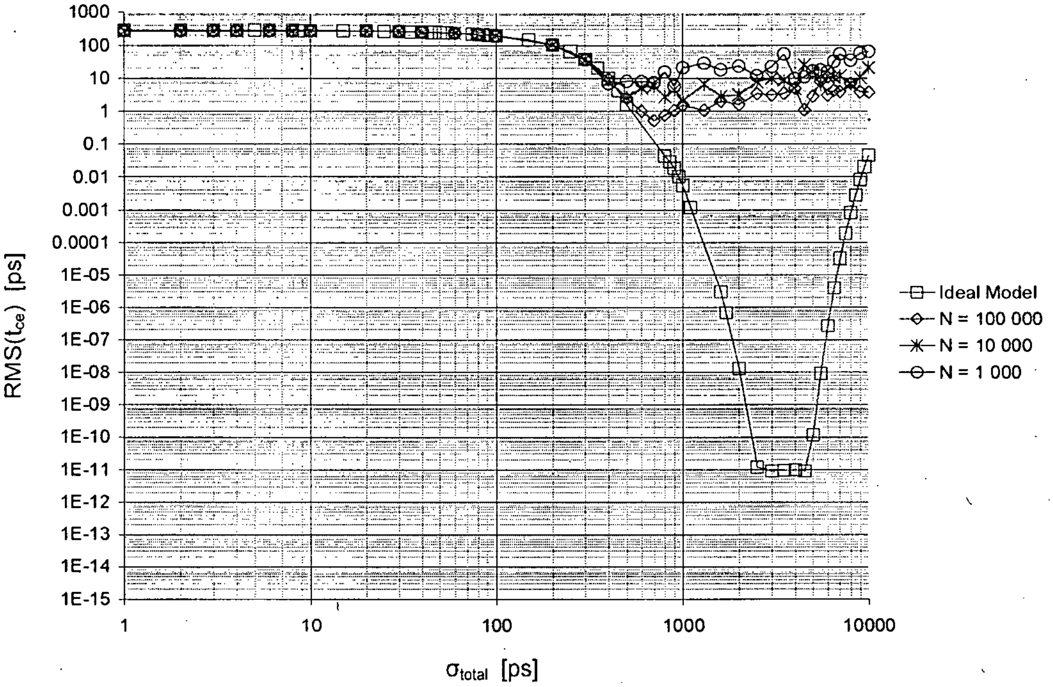


Figure 5.9: Comparison of the theoretical and realistic RMS error of the proposed calibration technique when $T_\Delta = 1$ ns.

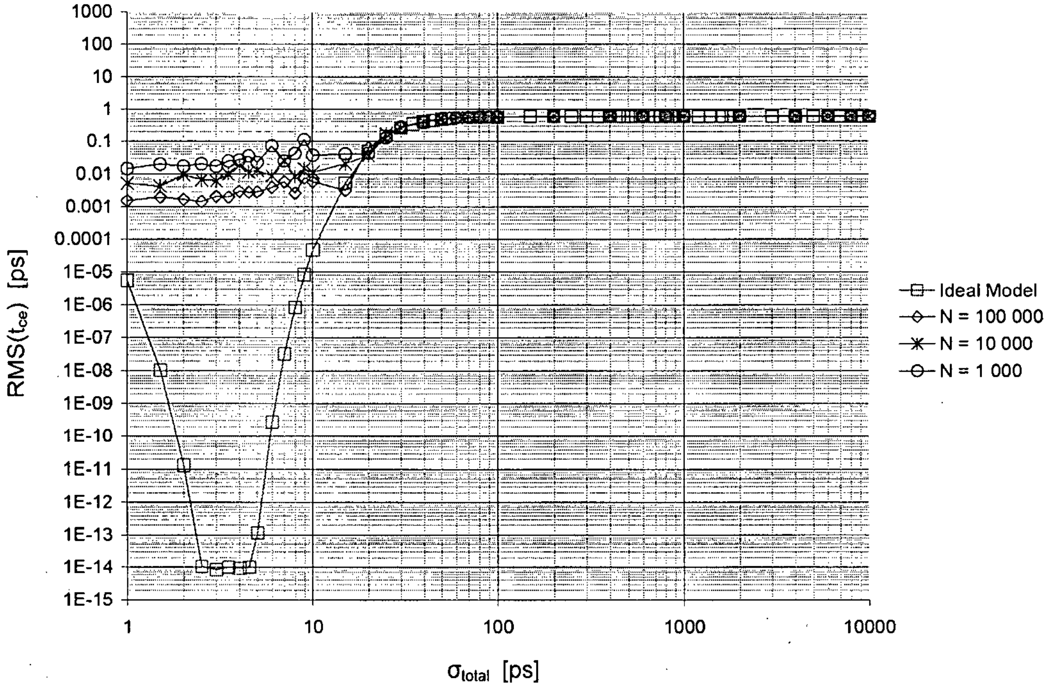


Figure 5.10: Comparison of the theoretical and realistic RMS error of the proposed calibration technique when $T_\Delta = 1$ ps.

Inspection of Figure 5.10 reveals that it is possible to achieve an RMS calibration error as low as 2 fs when 100 000 measurement repetitions are used. While calibration accuracies in the order of femtoseconds are certainly impressive, it is unclear what repercussions this level of accuracy will have on the required calibration time.

It is possible to calculate the calibration time for any value of T_Δ or N. The amount of time required by the proposed calibration technique to perform calibration is proportional to N and M, the number of measurement repetitions and the number of time intervals per measurement repetition, respectively, and inversely proportional to f_B, the frequency at which the time intervals are applied to the *START* and *STOP* inputs of an SOTDC. To obtain the total calibration time, this result must then be multiplied by a factor of two. This is true since the calibration procedure is not complete until the oscillator cycle counts have been captured using both the normal and the reversed orientations of the SOTDC's inputs. This relationship is summarized in Equation (5.5).

$$t_{cal} = \frac{2MN}{f_B} \tag{5.5}$$

While Equation (5.5) can be used to calculate the required calibration time of the proposed calibration technique, it is not useful when predicting the impact of a particular value of T_Δ on the total calibration time. A more convenient form of Equation (5.5) can be derived by recognizing that M, the number of time intervals per measurement repetition, is actually equivalent to Q, as shown in Equation (5.6).

$$M = \frac{T_A}{T_\Delta} = Q \tag{5.6}$$

As discussed in Section 4.3, Q is an integer chosen through careful selection of A and B, the PLL divisors, according to Equation (5.7).

$$Q = \frac{1}{\frac{A}{B} - 1} \tag{5.7}$$

Therefore, after substituting Equation (5.6) into Equation (5.5), a new calibration time expression is produced, as shown in Equation (5.8).

$$t_{cal} = \frac{2N}{f_A f_B T_\Delta} = \frac{2N}{f_A - f_B} = \frac{2QN}{f_B} \tag{5.8}$$

While any one of the three relationships of Equation (5.8) can be used to calculate the total calibration time, it is most sensible to use an expression that does not include T_Δ as a parameter. This is true since T_Δ is actually a function of the chosen calibration oscillator parameters, i.e., f_{in}, A, and B, and is therefore not independently selected. As previously mentioned, the calibration oscillators must produce a periodic sequence of time intervals, which can only be achieved through careful selection of the PLL divisors according to Equation (5.7). Once f_{in} and the PLL divisors have been chosen, T_Δ can be calculated according to Equation (5.9).

$$T_\Delta = \frac{1}{Q f_A} \tag{5.9}$$

If the chosen calibration oscillator parameters yield an unacceptable value of T_Δ for the given accuracy constraints, then one or all of f_{in}, A, and B must be reselected. Table 5.2 contains four sets of PLL divisors that can be used to produce four different T_Δ values, each differing from its predecessor by approximately a factor of ten.

Table 5.2: Four PLL divisors (A and B) and their corresponding T_Δ values (f_{in} = 50 kH).

A	B	Q	T_Δ [ps]
3335	2668	4	1 499.31
4191	4064	32	149.13
5397	5376	256	14.47
8196	8192	2048	1.19

Using the data contained in Table 5.2 it is possible to calculate the required calibration time of the proposed calibration technique over a range of T_Δ values spanning approximately 1.2 ps to 1.5 ns. Table 5.3 contains the resultant calibration times, with each T_Δ value calculated across three different measurement repetitions numbers (N).

Table 5.3: Time required to calibrate an SOTDC, assuming the PLL divisors of Table 5.2 (f_{in} = 50 kH) [s].

		T_Δ [ps]			
		1 499.3	149.1	14.5	1.2
N	1 000	5.98E-05	3.15E-04	1.90E-03	1.00E-02
	10 000	5.98E-04	3.15E-03	1.90E-02	1.00E-01
	100 000	5.98E-03	3.15E-02	1.90E-01	1.00E+00

With this information in hand it is now possible to determine the effect that varying the value of T_Δ has on both the calibration time and accuracy of the proposed calibration technique. To this end, the data contained in Table 5.3 has been plotted alongside a subset of the data illustrated in Figure 5.6 through Figure 5.10 inclusive.

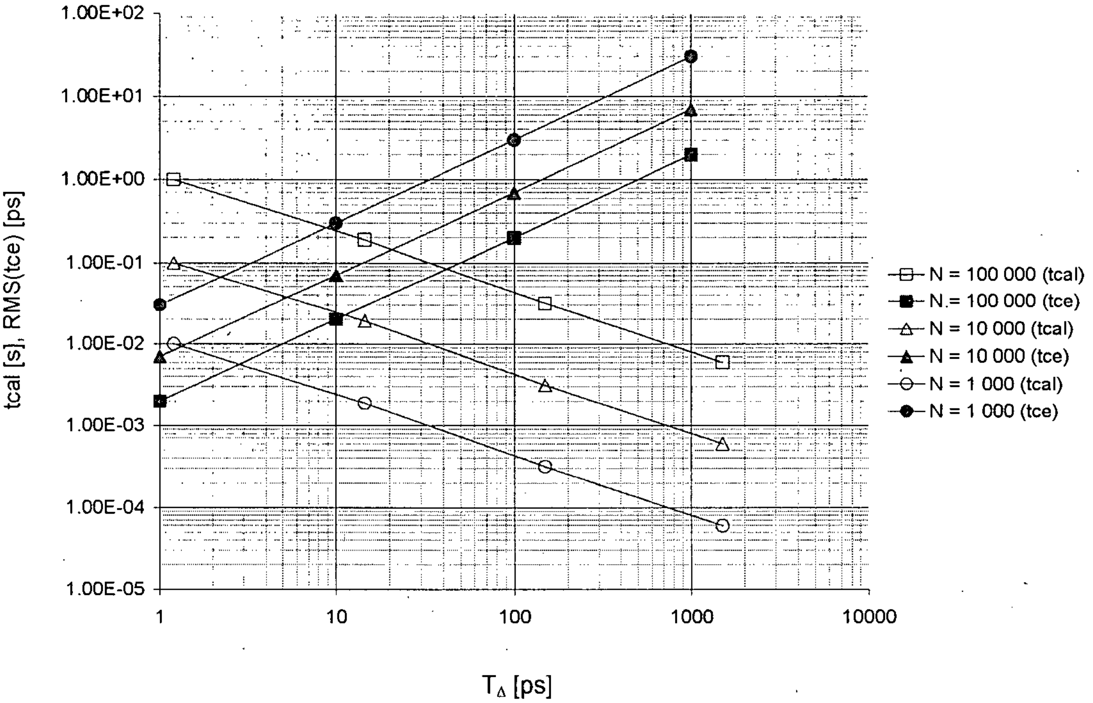


Figure 5.11: The effect of varying T_Δ on the calibration time (t_{cal}) and the calibration error (t_{ce}) (calculated using an optimal value of σ_{total}).

It is immediately evident by inspection of Figure 5.11 that the calibration error is adversely affected by an increase in the value of T_Δ, while just the opposite is true of the calibration time. While both observations are intuitive and have been supported either in theory or simulation, it is important to note that the trade-off which exists between calibration time and accuracy does have an intersection point, i.e., it is possible to assign an equal weighting to both calibration time and accuracy. However, since this intersection point is a function of N, the number of measurement repetitions, no universally optimal value of T_Δ exists. In fact, it can be observed from Figure 5.11 that the intersection points of the calibration time and error data sets decrease as N is lowered, while they also occur at lower values of T_Δ. Therefore, if both calibration time and error are of equal importance, then it is possible to minimize both by reducing the number of measurement repetitions employed during calibration in addition to appropriately reducing T_Δ.

While it is possible to reduce the calibration time and error of the proposed calibration technique by lowering both N and T_{Δ}, the latter may require careful consideration. For example, as shown in Table 5.2, both PLL divisors (A and B) must be increased in order to produce a lower value of T_{Δ}, assuming the PLL input frequency (f_{in}) remains constant. While the desired frequency synthesis can be achieved by increasing the depth of the counters in the PLL feedback path or by employing a cascaded PLL structure, a more serious issue of unwanted temporal noise may exist. For example, as T_{Δ} is decreased, the time intervals produced by the outputs of the two calibration oscillators could begin to deviate substantially from their intended distributions, depending upon the nature of the unwanted temporal noise. Therefore, there may be a practical limit to how far T_{Δ} can be lowered before unwanted temporal noise begins to limit the calibration accuracy of the proposed calibration technique.

Chapter 6

Conclusions and Future Work

As CMOS technology continues to advance, circuit timing problems are becoming more common and yet more difficult to diagnose. As a result, several sophisticated embedded time interval measurement techniques have been proposed to help address this growing problem. Perhaps the most promising of measurement techniques is the "sampling offset"-based flash time-to-digital converter (SOTDC). This embedded time interval measurement technique is capable of picosecond measurement accuracies in addition to millisecond test-times. However, the accuracy of an SOTDC is strongly dependent upon the capabilities of its calibration technique, and present SOTDC calibration techniques suffer from some very serious limitations. In fact, these limitations are so severe that present calibration techniques are impractical under realistic production test conditions.

6.1 Summary and Contributions

In order to address the need for a feasible and accurate embedded SOTDC calibration technique, a new calibration technique has been proposed. This technique leverages the advantages of the added noise-based calibration technique, while doing away with its limitations. The proposed calibration technique's method of operation can be described as follows:

1) Two PLLs, each with carefully chosen divisors, are locked to a single known reference frequency and used to generate a periodic sequence of time intervals.
2) The output of one PLL is modulated such that its period distribution conforms to a Gaussian PDF, whose standard deviation has been appropriately selected.
3) A counter is then used to store the number of PLL clock cycles that are elapsed between the switching-events of a reference arbiter and the arbiter under calibration. Several thousand measurements are accumulated by the counter in order to increase the accuracy of the estimated arbiter sampling offset.
4) The PLL divisors are then swapped in order to effectively reverse the inputs to the arbiter under calibration, and the counter is once again used to store the number of PLL clock cycles that are elapsed between the switching-events of a reference arbiter and the arbiter under calibration.
5) The sampling offset of the arbiter under calibration is then estimated by performing simple mathematical operations on the captured counter values.

The main contribution of the proposed calibration technique is the ability to perform calibration without knowledge of the values of the time intervals applied to the SOTDC during calibration. Only knowledge of the temporal difference between adjacent time intervals is required. This information can be acquired through the selection of a reference frequency and PLL divisors according to Equation (4.34). In addition, there is no need to apply a curve fitting function to the calibration results. Post-processing of the results consists of simple subtraction, multiplication, and division operations. Therefore, the post-processing requirements of the proposed calibration technique are much less demanding than those of either the direct or the added noise-based calibration techniques.

In order to understand the capabilities of the proposed calibration technique, a set of statistically-based mathematical equations were derived. These equations were used to predict the estimated sampling offset of an arbiter calibrated using the proposed SOTDC calibration technique, and thus to determine the theoretical accuracy of this technique. These results were then compared to those of a behavioural Matlab model of the proposed calibration technique. It was concluded that the accuracy of the proposed technique is determined by the number of measurement repetitions performed during calibration, assuming a fixed value of T_Δ and an appropriate value of σ_{total}. Finally, it was shown that it is possible to obtain both sub-picosecond calibration accuracies and sub-100 millisecond calibration times, while still placing realistic demands on the time intervals used during calibration, and hence the calibration oscillators. Therefore, the desirable features of the added noise-based calibration technique, i.e., sub-picosecond calibration accuracies using realistic time interval resolutions, have been maintained; however the impractical implementation requirements of such a technique have been eliminated.

6.2 Future Work

While the SOTDC calibration technique proposed in this thesis has been discussed in some detail, the opportunity for further investigation remains possible. The following subsections present three avenues of future work.

6.2.1 The impact of non-idealities

While it has been assumed over the course of this thesis that thermal noise is the only source of noise in an arbiter, in realty this is not the case. An arbiter fabricated in a modern CMOS process is subjected to several different sources of noise, each of which need to be considered in order to truly understand the potential of the proposed calibration technique. For example, power supply and substrate noise may alter the sampling offset of an arbiter in an unpredictable manner. However, it remains to be seen whether the resulting sampling offset variation would be significant enough to impact the recorded number of oscillator cycle counts, and hence the accuracy of the proposed technique. This is true since the temporal noise that is added to each time interval, or equivalently to the arbiters themselves, may be so large as to dominate over the undesirable noise sources.

Similarly, it has been assumed during the course of this thesis that it is possible to generate a periodic sequence of time intervals by locking two PLLs to a stable reference frequency. However, as shown in Table 5.2, the divisors employed in each PLL may be in the order of several thousand. While it is possible to achieve the desired frequency synthesis by employing a cascaded PLL structure, the resultant output signals may contain a significant amount of unwanted phase noise. Again, the effect of the undesired phase noise may depend upon the extent to which the time intervals are intentionally modulated. However, in order to understand the potential of the proposed calibration technique, the limitations imposed by PLL phase noise must be identified.

6.2.2 Circuit implementation

In order to prove the viability of the proposed calibration technique, a working implementation must first be demonstrated. One possible embodiment of the proposed calibration technique can be found in Appendix A, along with the schematics of a 16-bit SOTDC circuit. As shown in Appendix A, the output of each arbiter, including the reference arbiter, is sampled using a delayed version of one of the calibration oscillators (PLLs). The reference arbiter is sampled in order to detect its switching-event, which is then used to trigger the 16 24-bit oscillator cycle counters of the arbiters under calibration. However, an additional counter is required in order to determine the correct switching-event of the reference arbiter, as the addition of temporal noise to the time intervals can induce a false reference arbiter switching-event. Therefore, a counter is used as a shift-register to store the sampled output of the reference arbiter over 24 cycles. This data can then be used to determine the number of oscillator cycles for which the reference arbiter output has been sampled as logic ‘0’.

In theory, as the number of consecutive ‘0’s in the shift-register grows, it becomes increasingly likely for a valid reference arbiter switching-event to occur. However, this hypothesis has not been proven mathematically or experimentally, and is therefore a candidate for further exploration. Similarly, it should be demonstrated experimentally that a reference arbiter with a constant sampling offset, irrespective of the orientation of its inputs, can be constructed from two over-sized symmetric CMOS arbiters with matched layouts.

6.2.3 Additional applications and SOTDC improvements

While the primary focus of this thesis has been the calibration of an SOTDC, the proposed calibration technique may be useful in a wide variety of applications. For example, the proposed calibration technique may be used to accurately calibrate the delay of a variable delay line, instead of the sampling offset of a symmetric CMOS arbiter. Analogously, the proposed calibration technique may be used to measure the delay of an inverter driving a known load. Such information is often helpful when estimating the strength of a CMOS process, which in turn can be used to calibrate the bias currents of analog circuitry on a shared die [18].

Lastly, a conventional SOTDC, such as the one presented in this thesis, is characterized by a limited dynamic range, i.e., the ratio of the maximum to the minimum measurable time interval is generally less than one hundred. However, it may be possible to greatly extend the dynamic range of an SOTDC, and therefore to increase its scope. For example, instead of using a single SOTDC to measure a long time interval, it is conceivable that two SOTDCs could be used to divide the time interval into smaller, more manageable units; i.e., ones that do not exceed the dynamic range of either SOTDC. Under such a scenario, it would then be possible to measure the entire time interval by alternating between the two SOTDCs when measuring the reduced time intervals. The creation of the reduced time intervals may be achieved by pre-empting the *STOP* signal of the active SOTDC before its dynamic range has been exceeded, while simultaneously initiating the measurement of a new time interval on the alternate SOTDC by triggering its *START* signal. Once this process has terminated, the counters within the two SOTDCs may be analysed in order to estimate the value of the entire time interval. However, this proposal has not been formally investigated, and as such is a candidate for future examination.

References

[1] D. A. Hodges, H. G. Jackson, and R. A. Saleh, *Analysis and Design of Digital Integrated Circuits*, McGraw-Hill, 2003, ch. 10, 11.

[2] M. Xu, "*Substrate Noise in Mixed-Signal Integrated Circuits,*" Ph.D. thesis, Center for Integrated Systems, Stanford University, Stanford, CA, USA, 2001, ch 1, 2, 3.

[3] K. Mollah, "*Design of a Tunable CML-based Differential Ring Oscillator with Short Start-up and Switching Transients,*" M.A.Sc. thesis, Dept. of Electrical and Computer Engineering, University of British Columbia, Vancouver, BC, Canada, 2004, ch 2.

[4] *Jitter Fundamentals*, WAVECREST Corporation, 2001 [Online], Available: http://www.wavecrest.com/technical/pdf/jittfun_hires_sngls.pdf

[5] *Jitter Measurements for CLK Generators or Synthesizers*, Maxim Integrated Products, Application Note, 2003 [Online], Available: http://www.maxim-ic.com/appnotes.cfm/appnote_number/2744

[6] *Jitter Measurements in Serial Data Signals*, LeCroy Corporation, White Paper, 2007 [Online], Available: http://www.lecroy.com/tm/Library/WhitePapers/PDF/WP_JitterMeasurement_in_SerialDataSignals.pdf

[7] B. Analui, J. F. Buckwalter, and A. Hajimiri, "Data-Dependent Jitter in Serial Communications," *IEEE Transactions on Microwave Theory and Techniques*, vol. 53, no. 11, November 2005.

[8] *Three Views Of Jitter*, LeCroy Corporation, 2002 [Online], Available: http://www.lecroy.com/tm/options/software/jitter-and-JTA/page04.asp

[9] J. Kalisz, "Review of methods for time interval measurements with picosecond resolution," *Institute of Physics Publishing*, Metrologia 41, pp. 17–32, 2004.

[10] M. Li, "Requirements, Challenges, And Solutions For Testing Multiple GB/s ICs In Production," *ITC International Test Conference*, Panel P8.3, pp. 1309, 2003.

[11] Y. Zheng and K. L. Shepard, "On-chip oscilloscopes for non-invasive time-domain measurements of waveforms in digital integrated circuits," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 11, no. 3, pp. 336-344, June 2003.

[12] T. Yamaguchi and M. Soma, "A Method for Measuring the Cycle-to-Cycle Period Jitter of High-Frequency Clock Signals," *Proc. 19th IEEE VLSI Test Symp. (VTS 01)*, pp. 102-110, 2001.

[13] H. Lane, "Improving Test Times with Fast Frequency, Amplitude and Waveform Switching,“ *High Frequency Electronics*, pp 32-37, November 2006.

[14] *SIA 3600 – Signal Integrity Analysis Solution*, WAVECREST Corporation, Application Note, 2003 [Online], Available: http://www.wavecrest.com/products/datasheet/SIA3600datasheet.pdf

[15] *High-Speed Serial Interface Test in Production*, GuideTech Inc., 2004 [Online], Available: http://www.guidetech.com/pdf/wp/LearnCTIA/ITCGuideTechExhibitorPresentation-Oct2004finalgc.ppt

[16] *IP/Network Measuring Instruments*, Anritsu Corporation, 2006 [Online], Available: http://www.anritsu.co.jp/Products/pdf_e/B05E_MP1590B.pdf

[17] T. M. Mak, M. Tripp, and A. Meixner, "Testing Gbps Interfaces without a Gigahertz Tester," *IEEE Design & Test of Computers*, July-August 2004, pp. 278-286.

[18] R. B. Staszewski, S. Vemulapalli, P. Vallur, J. Wallberg, and P. T. Balsara, "1.3V 20ps Time-to-Digital Converter for Frequency Synthesis in 90-nm CMOS," *IEEE Transactions on Circuits and Systems II*, vol. 53, no. 3, pp. 220-224, March 2006.

[19] S. Tabatabaei, "*Embedded Test Circuits and Methodologies for Mixed-Signal ICs,*" Ph.D. thesis, Dept. of Electrical and Computer Engineering, University of British Columbia, Vancouver, BC, Canada, 2000.

[20] S. Sunter and A. Roy, "On-chip digital jitter measurement, from megahertz to gigahertz," *IEEE Design & Test of Computers*, July-August 2004, pp. 314-321.

[21] H. Dang, M. Sawan, Y. Savaria, "A novel approach for implementing ultra-high speed flash ADC using MCML circuits," *IEEE International Symposium on Circuits and Systems*, vol. 6, pp. 6158-6161, May 2005.

[22] D. M. Santos, "A CMOS delay locked loop and sub-nanosecond time-to-digital converter chip," *IEEE Transactions on Nuclear Science*, vol. 43, pp. 1717-1719, June 1996.

[23] V. Gutnik, "*Analysis and characterization of random skew and jitter in a novel clock network*," Ph.D. thesis, MIT, Cambridge, MA, USA, pp. 83-92, June 2000.

[24] P. Levine, "*High-Resolution Time Measurement and Calibration for On-Chip Test Systems,*" M.A.Sc. thesis, Dept. of Electrical and Computer Engineering, McGill University, Montreal, QC, Canada, 2004, ch 4.

[25] V. Gutnik and A. Chandrakasan, "Multiple Arbiter Jitter Estimation System and Related Techniques," US Patent no. 6,661,860, Patent and Trademark Office, Washington DC, 2003.

[26] P. M. Levine and G. W. Roberts, "A high-resolution flash time-to-digital converter and calibration scheme," *ITC International Test Conference*, 2004, pp. 1148-1157.

[27] L. Kleeman, "The jitter model for metastability and its application to redundant synchronizers," *IEEE Transactions on Computers,* vol. 39, no. 7, pp. 930-942, July 1990.

[28] W. A. M. Van Noije, W. T. Liu, and S. J. Navarro, "Precise final state determination in mismatched CMOS latches," *IEEE Journal of Solid State Circuits*, vol. 30, no. 5, pp. 607-611, May 1995.

[29] W. Lian and X. S. Cheng, "Analytical model of noise of a switched flip-flop," *Electronics Letters*, vol. 24, no. 21, pp. 1317-1318, October 1988.

[30] *Erf,* Wolfram MathWorld, 2007 [Online], Available: http://mathworld.wolfram.com/Erf.html

[31] *Root-Mean-Square*, Wolfram MathWorld, 2007 [Online], Available: http://mathworld.wolfram.com/Root-Mean-Square.html

[32] A. Hajimiri, T. H. Lee, *The Design of Low Noise Oscillators*, Kluwer Academic Publishers, 1999, ch. 2, 3, 5.

[33] J. A. McNEILL, "*Jitter In Ring Oscillators,*" Ph.D. thesis, College of Engineering, Boston University, Boston, MA, USA, 1994.

[34] B. Nelson and M. Soma, "On-chip calibration technique for delay line based BIST jitter measurement," *Proceedings of the 2004 International Symposium on Circuits and Systems*, pp. I-944 – I-947, May 2004.

[35] C. M. Grinstead and J. L. Snell, *Introduction to Probability*, American Mathematical Society, 1997.

[36] *Independent Statistics*, Wolfram MathWorld, 2007 [Online], Available: http://mathworld.wolfram.com/IndependentStatistics.html

[37] G. Evans, J. Goes, A. Steiger-Garção, M. D. Ortigueira, N. Paulino and J. Sousa Lopes, "Low-Voltage Low-Power CMOS Analogue Circuits for Gaussian and Uniform Noise Generation," *Proceedings of the 2003 International Symposium on Circuits and Systems*, Volume 1, pp. I-145 - I-148, May 2003.

[38] P. Dudek, S. Szczepanski and J. Hatfield, "A high-resolution CMOS time-to-digital converter utilizing a Vernier delay line," *IEEE Journal of Solid-State Circuits*, vol. 35, no. 2, pp. 240-247, February 2000.

[39] *Normal Distribution*, Wolfram MathWorld, 2007 [Online], Available: http://mathworld.wolfram.com/NormalDistribution.html

Appendix A

Circuit Implementation of Proposed Calibration Technique and 16-bit SOTDC

This appendix contains the schematics of a conventional 16-bit SOTDC circuit, in addition to those of the proposed calibration technique.

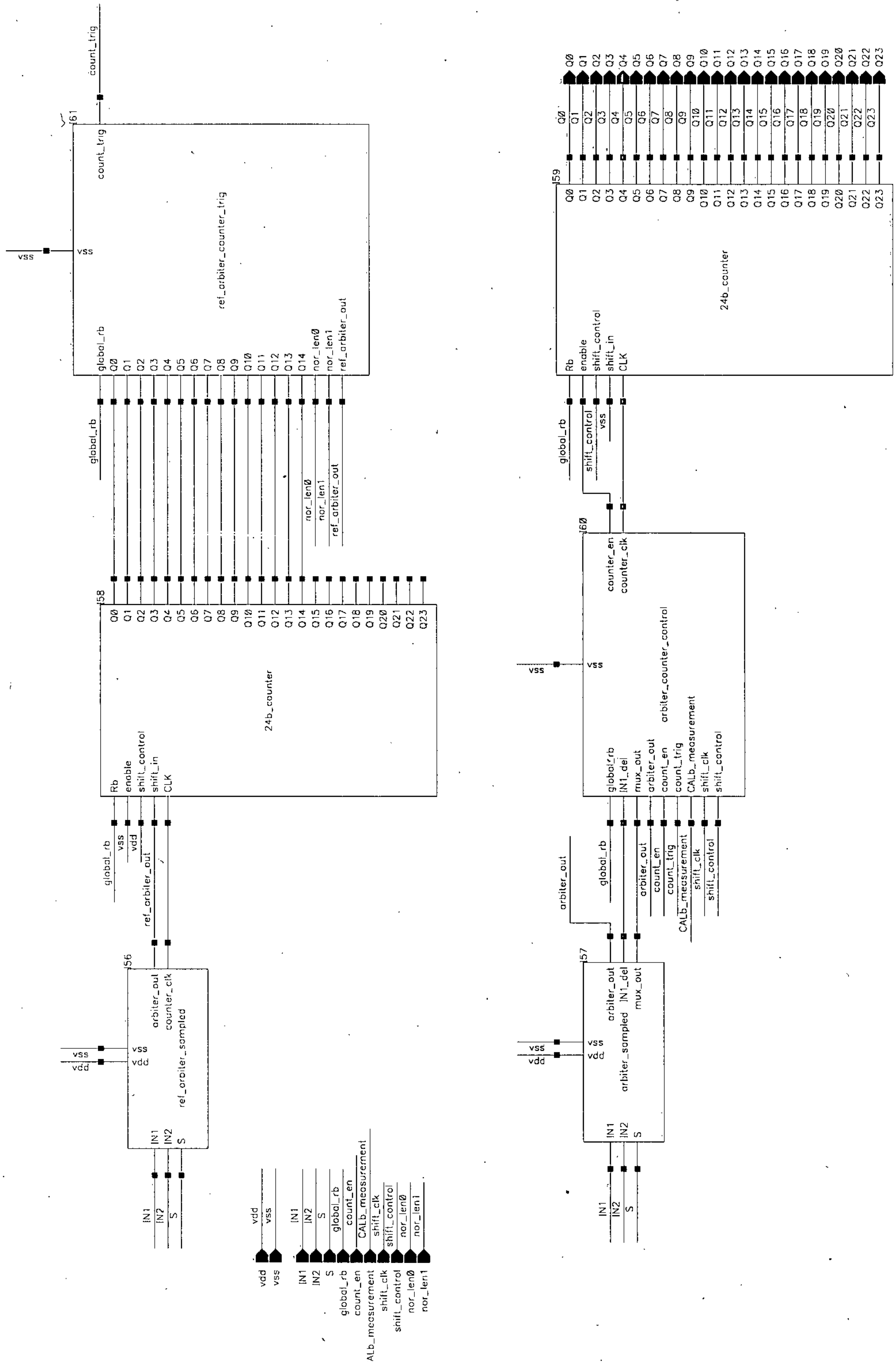


Figure A.1: Top-level schematic of the proposed SOTDC calibration circuit (1-bit of 16-bit SOTDC illustrated).

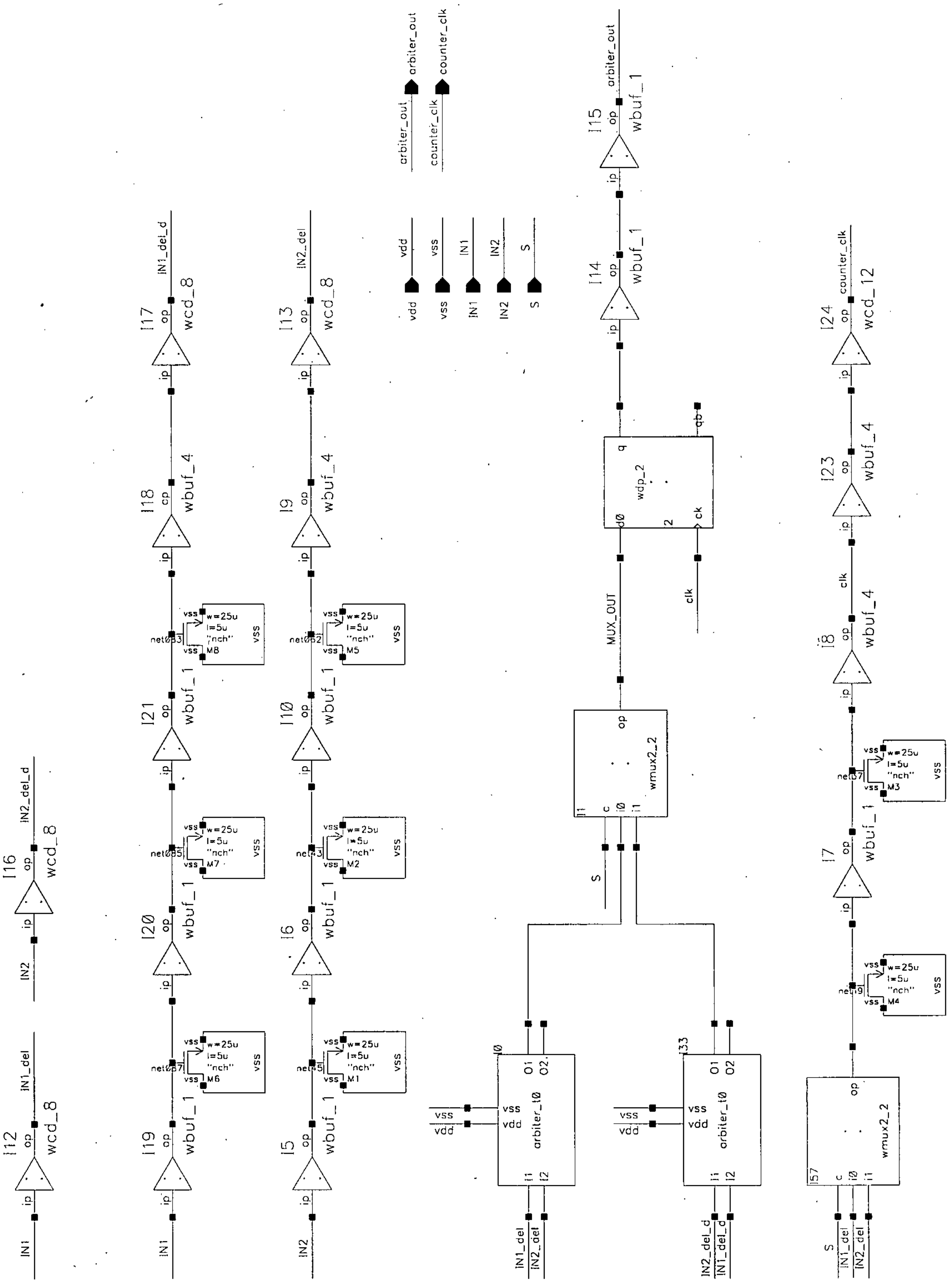


Figure A.2: Schematic of the Reference Arbiter Sampling circuit

("ref_arbiter_sampled").

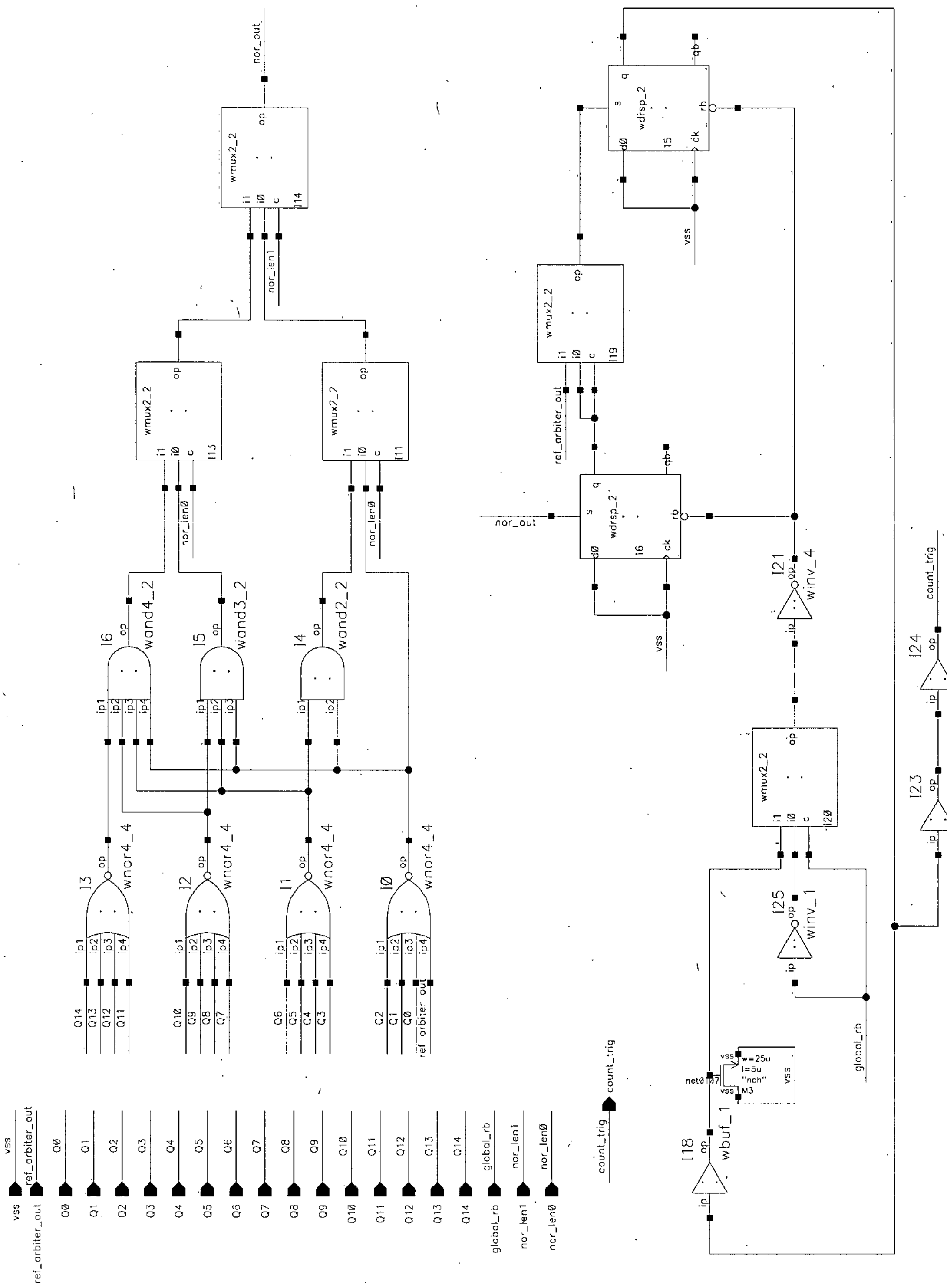


Figure A.3: Schematic of the Arbiter Array Counter Trigger circuit

("ref_arbiter_counter_trig").

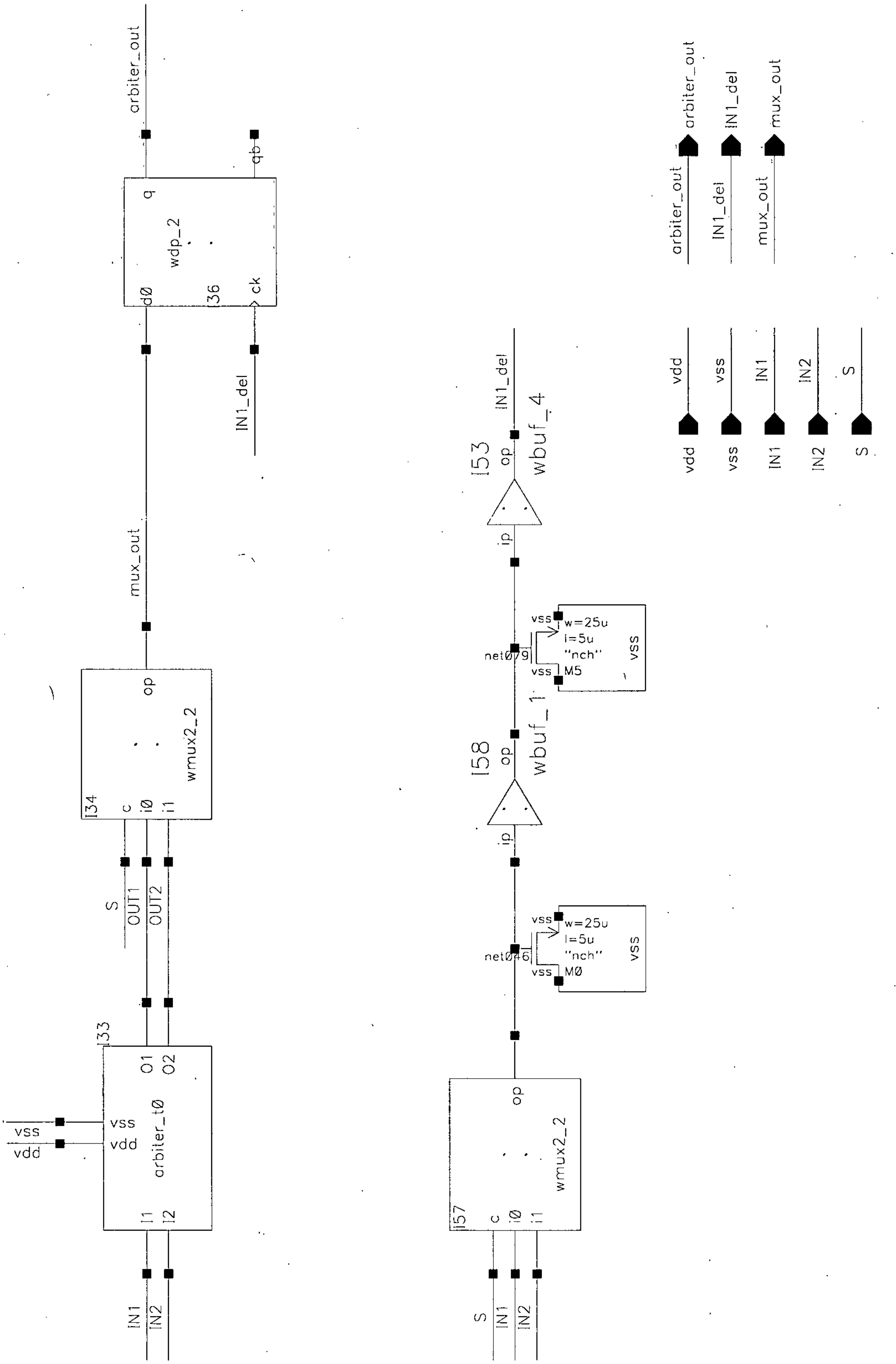


Figure A.4: Schematic of the Arbiter Sampling circuit ("arbiter_sampled").

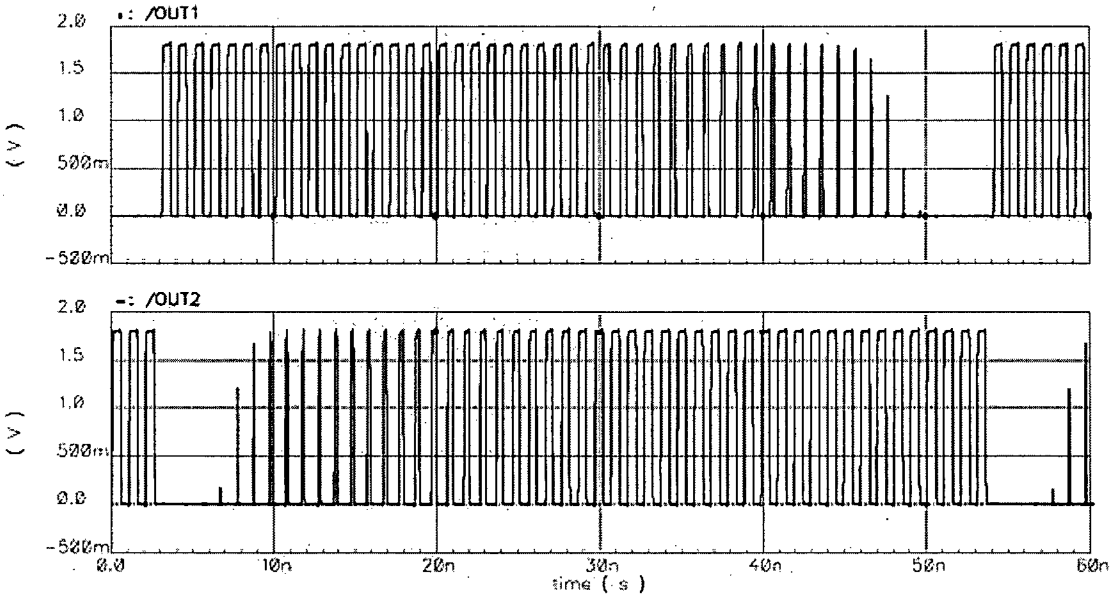


Figure A.5: The output of a positively biased arbiter when two free-running oscillators of different frequency, *oscA* and *oscB*, are applied to its inputs (T_A = 1 ns, T_Δ = 20 ps).

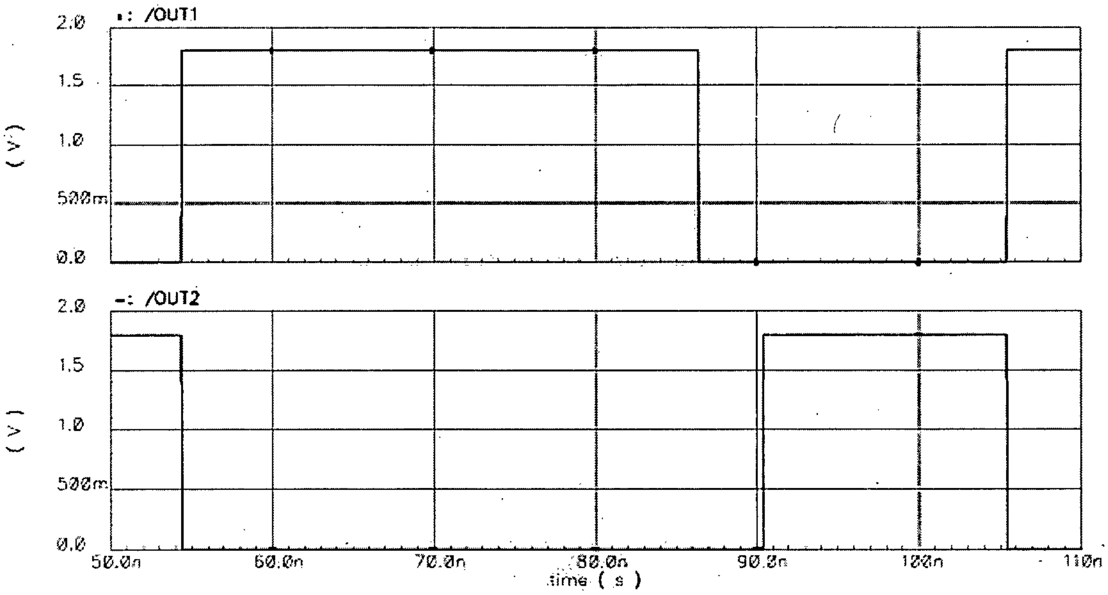


Figure A.6: The sampled output of a positively biased arbiter when two free-running oscillators of different frequency, oscA and oscB, are applied to its inputs (T_A = 1 ns, T_Δ = 20 ps, sampling delay = 50 ps).

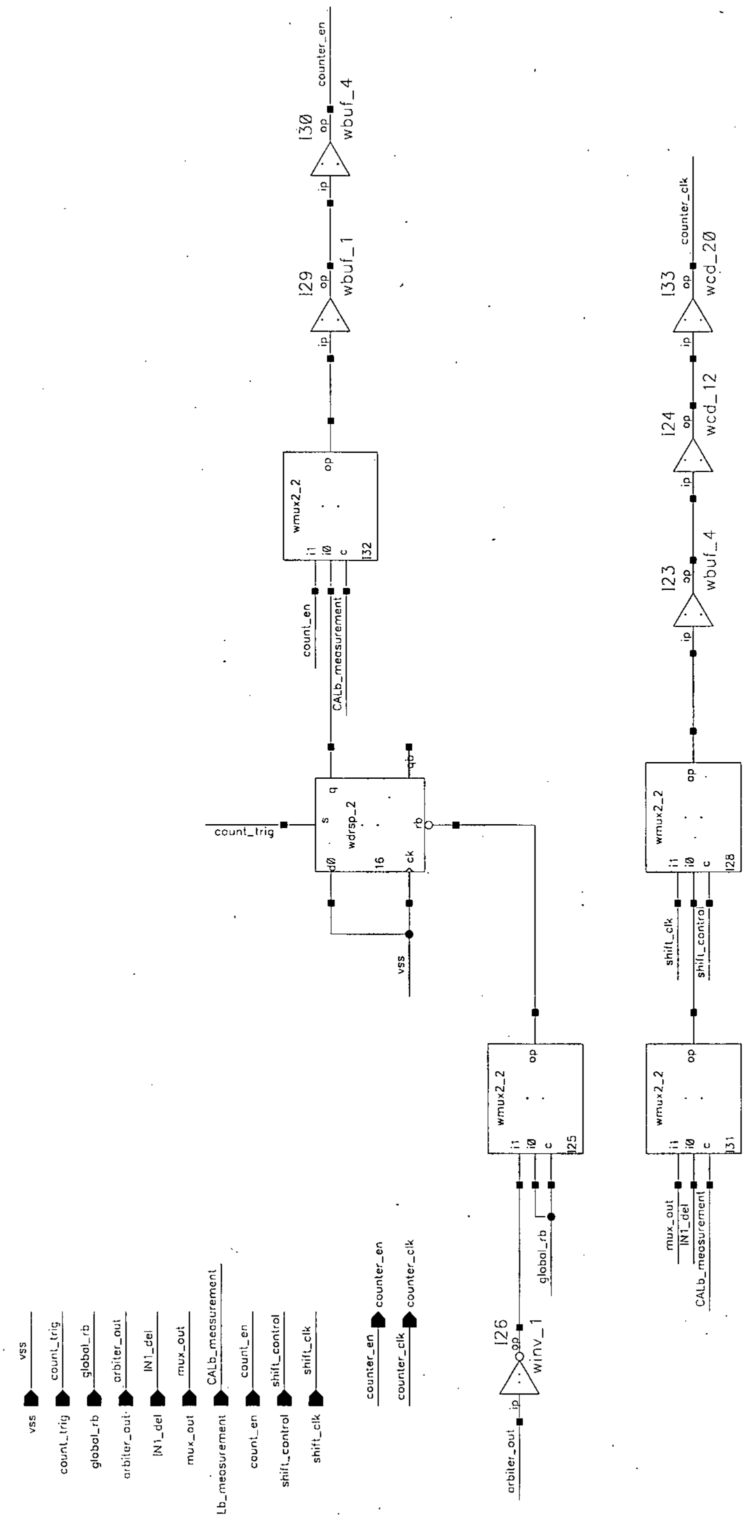


Figure A.7: Schematic of the Arbiter Counter Control circuit ("arbiter_counter_control").

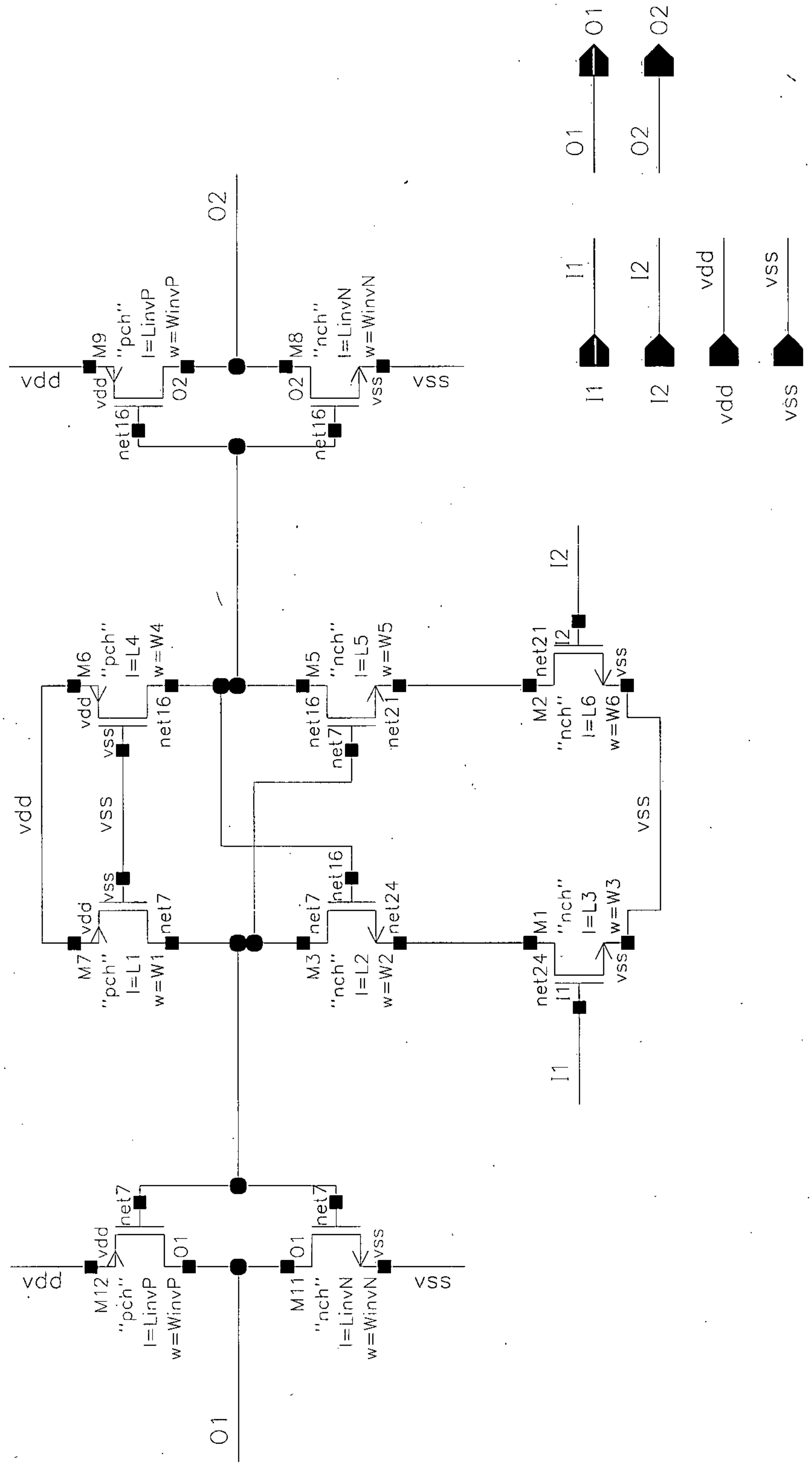


Figure A.8: Schematic of the Arbiter circuit ("arbiter_t0").

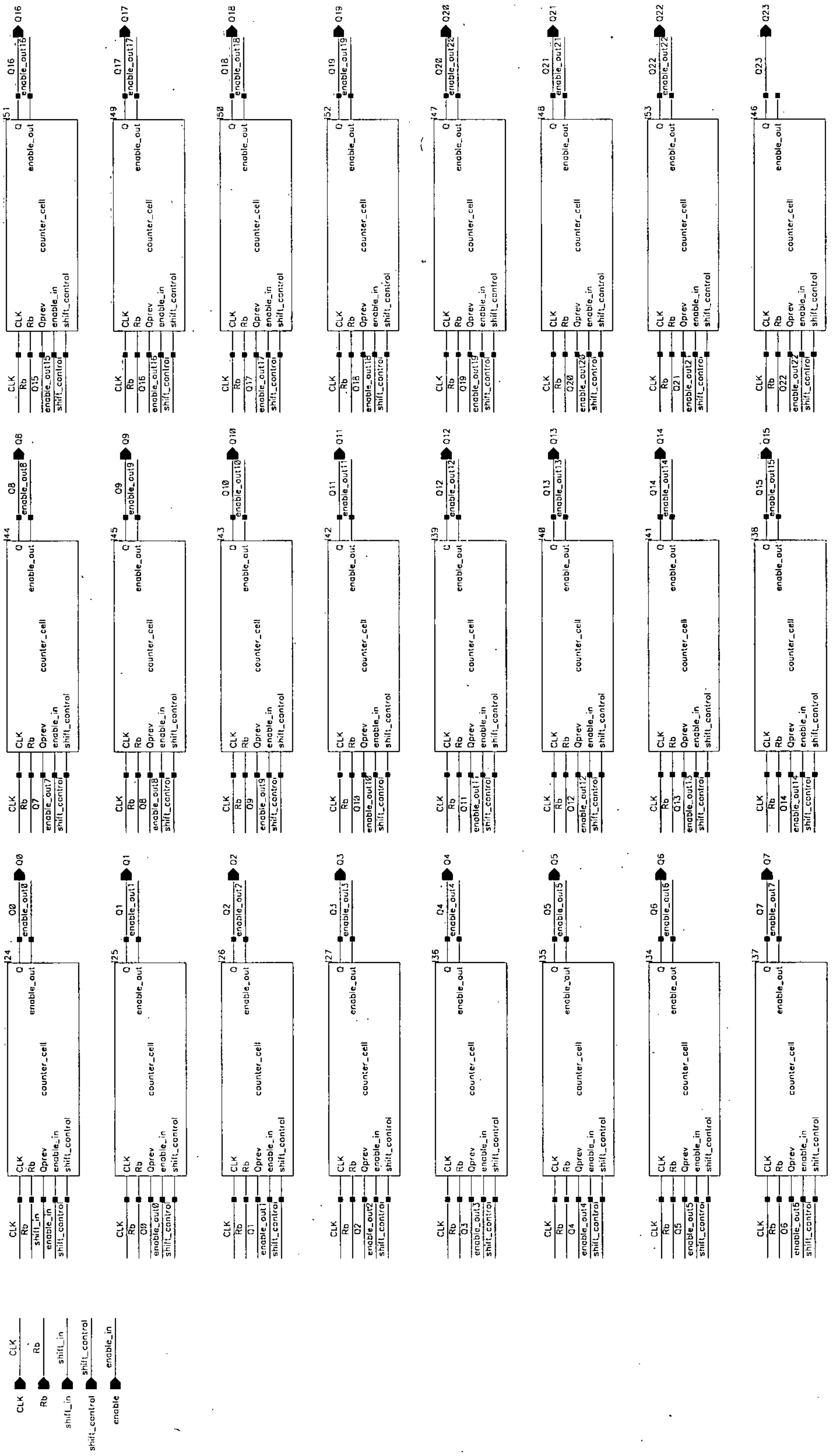


Figure A.9: Schematic of the 24-bit Counter circuit ("24b_counter").

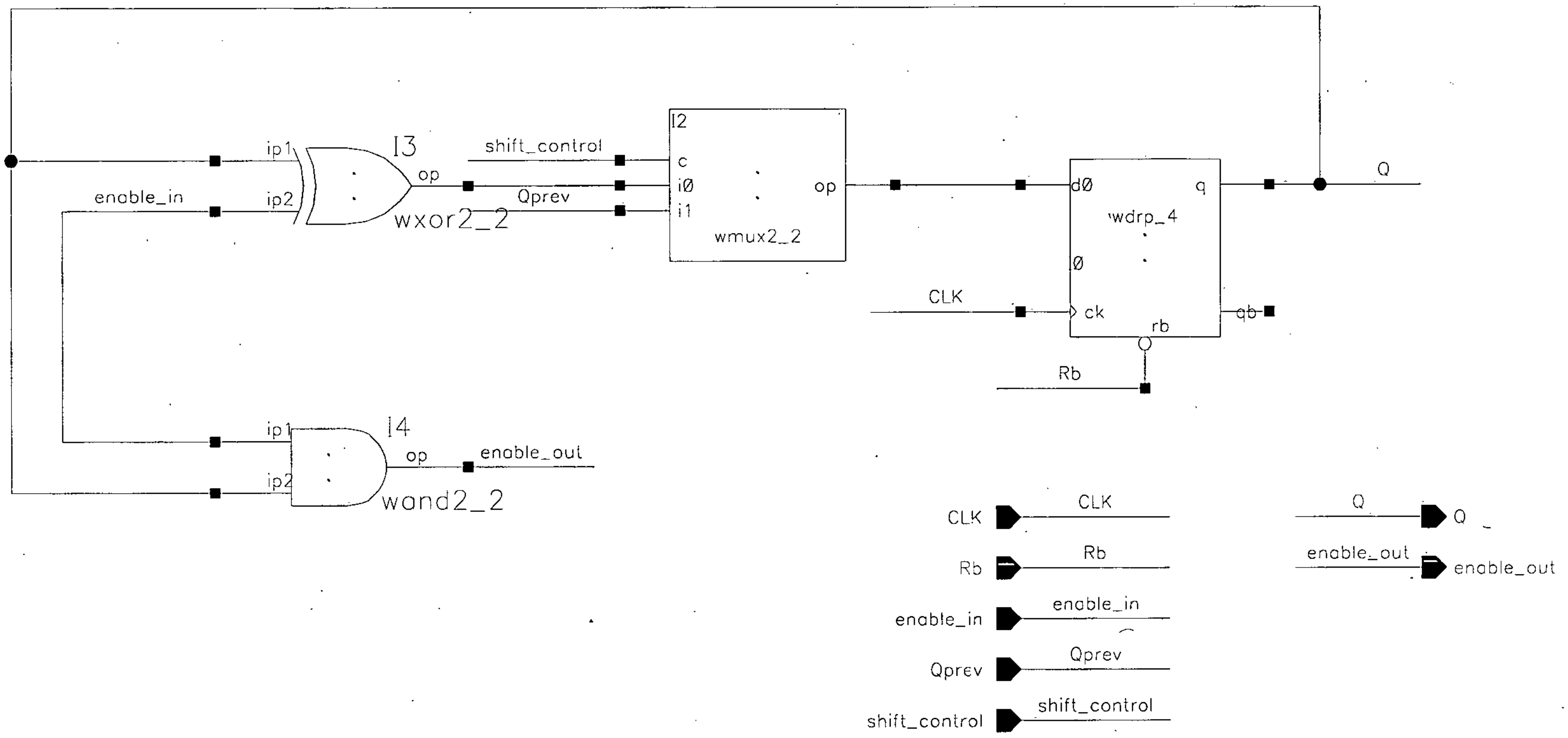


Figure A.10: Schematic of one stage of a 24-bit Counter (“counter_cell”).

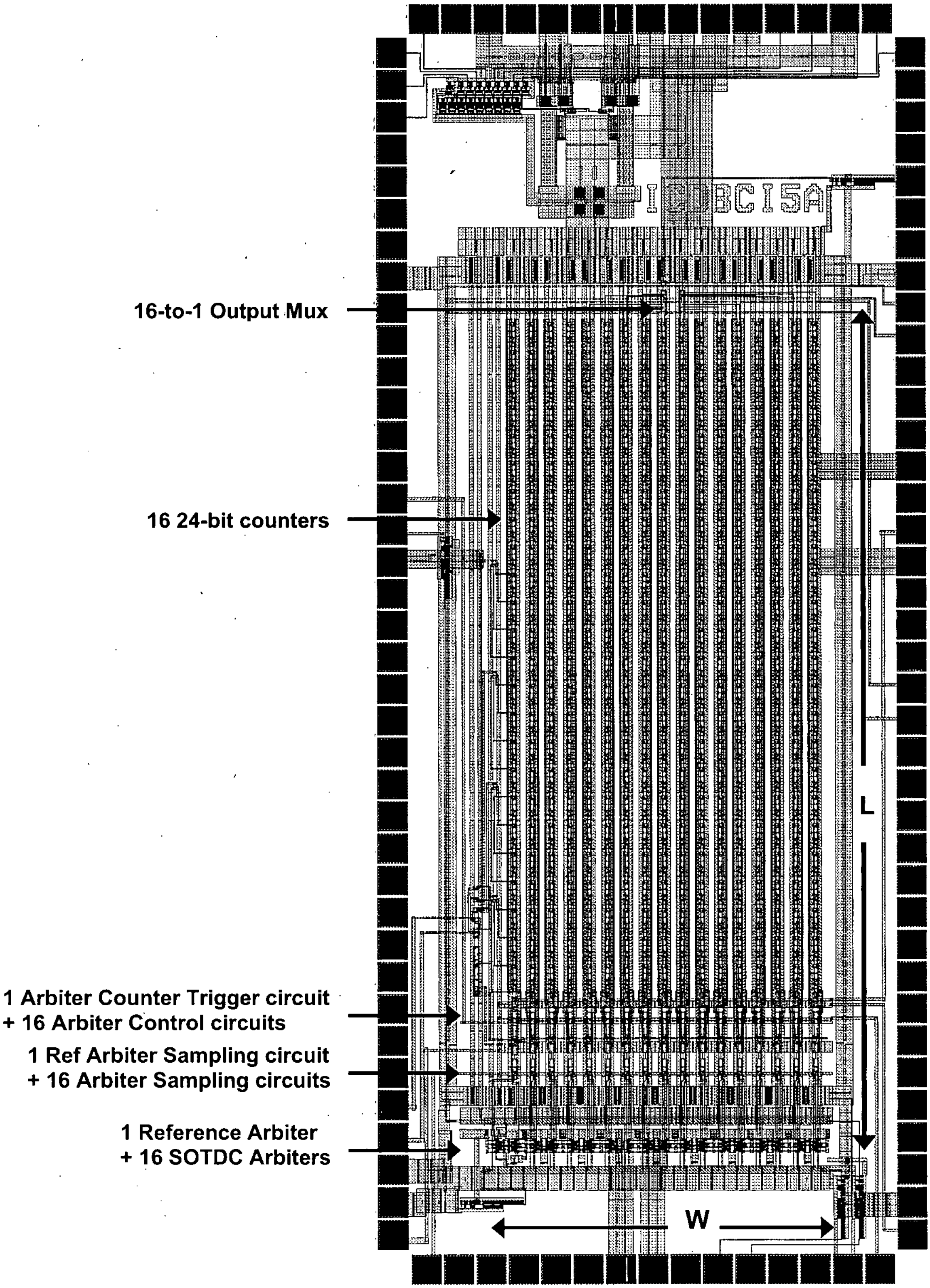

16-to-1 Output Mux
16 24-bit counters
1 Arbiter Counter Trigger circuit
+ 16 Arbiter Control circuits
1 Ref Arbiter Sampling circuit
+ 16 Arbiter Sampling circuits
1 Reference Arbiter
+ 16 SOTDC Arbiters
L
W

Figure A.11: Layout view of a 16-bit SOTDC and proposed calibration circuit in 0.35 um CMOS (L = 1930 um, W = 690 um).